- Highly Accelerated Stress Screens (HASS) (Equipment-level)
- Highly Accelerated Temperature and Humidity Stress Test (HAST) (Part-level)

Highly accelerated testing is the systematic application of environmental stimuli at levels well beyond those anticipated during product use. Thus, the results need to be carefully interpreted. It is used to identify relevant faults and to assure that the resulting products have a sufficient margin of strength above that required to survive the normal operating environments. Highly accelerated testing attempts to greatly reduce the time needed to precipitate these defects. The approach may be used either for development testing or for screening.

HALT is a development tool and HASS is a screening tool. They are frequently employed in conjunction with one another. They are new, and are in conflict with the classical approach to accelerated testing; thus, they are controversial. Their specific goal, however, is to improve the product design to a point where manufacturing variations and environment effects have minimal impact on performance and reliability. There is usually no quantitative life or reliability prediction associated with highly accelerated testing.

8.7.6.1 Step Stress Profile Testing

Using a step stress profile, test specimens are subjected to a given level of stress for a preset period of time, then they are subjected to a higher level of stress for a subsequent period of time. The process continues at ever increasing levels of stress, until either; all specimens fail, or the time period at the maximum level stress ends, as shown in Figure 8.7-2. This approach provides more rapid failures for analysis, but with this technique it is very difficult to properly model the acceleration and hence to quantitatively predict the item life under normal usage.

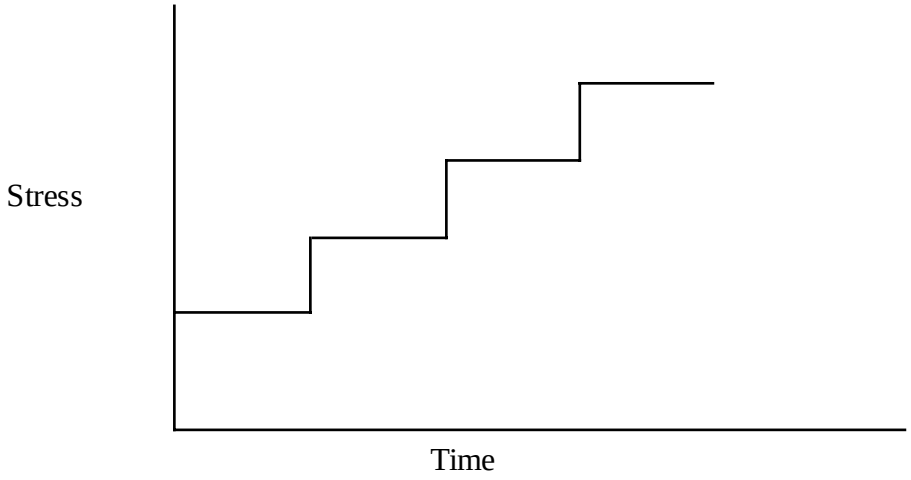


FIGURE 8.7-2: STEP STRESS PROFILE

How much to increase the stress in any single step is a function of many variables and is beyond the scope of this discussion. However, the general rule to follow in the design of such a test is to eventually exceed the expected environments by a comfortable margin so that all members of the population can be expected to survive both the field environment and the screen environments, assuming of course that they are defect free.

8.7.6.2 Progressive Stress Profile Testing

A progressive stress profile or "ramp test" is another frequently used approach (see Figure 8.7-3). With this approach the level of stress is continuously increased with time. The advantages and disadvantages are the same as those for step stress testing, but with the additional difficulty of accurately controlling the rate of increase, of the stress.

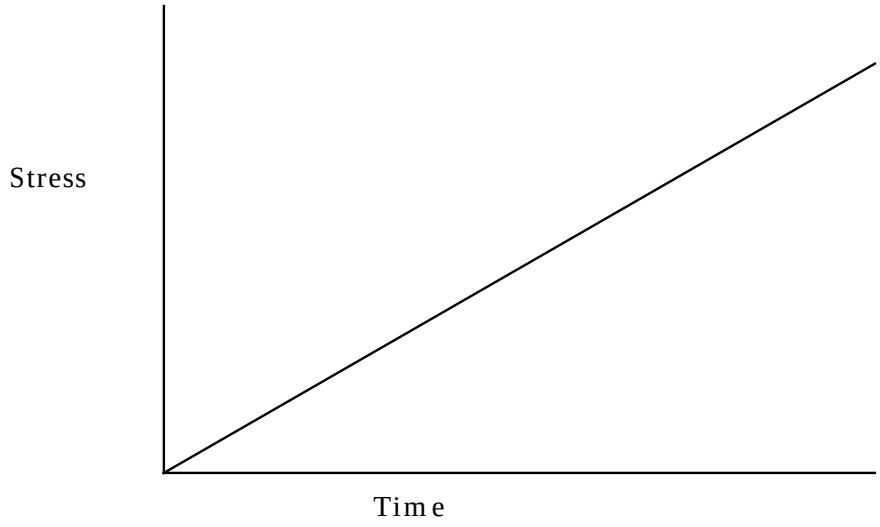


FIGURE 8.7-3: PROGRESSIVE STRESS PROFILE

8.7.6.3 HALT Testing

The term HALT was coined in 1988 by Gregg K. Hobbs (Ref. [8]). HALT (also, sometimes referred to as STRIFE (Stress plus Life) testing) is a development test, an enhanced form of step stress testing. It is typically used to identify design weaknesses and manufacturing process problems and to increase the margin of strength of the design rather than to predict quantitative life or reliability of the product.

HALT testing begins with step stress testing in generic stresses such as temperature, rate of change of temperature, vibration, voltage, power cycling and humidity. In addition, product unique stresses such as clock frequency, DC voltage variation and even component value variation may be the accelerated stimuli.

The type of the vibration stimuli used for HALT (and HASS) testing is unique. It is not based upon the universally accepted accelerated (power) spectral density concept. Thus it does not utilize classical, single-axis, sinusoidal vibration or a random vibration spectrum, generated by acceleration-controlled electro-dynamic shakers. Instead an unconventional multi-axial pneumatic (six degree of freedom) impact exciter is typically used. This type of equipment generates a highly unique broadband accelerated shock response spectrum (SRS). This is effectively a repeated shock environment rather than a vibration environment and is, in its self, much more severe than a classical vibration spectrum. Because of the choice of this shock stimuli spectrum, the resulting data cannot be easily correlated with either: (a) the normal environment or with (b) classical vibration testing using classical vibration modeling approaches. Thus quantitative prediction of life or reliability is not usually possible with HALT and HASS.

Using HALT the step stress process continues until stress levels well above those expected in the normal operational environments are exceeded. Throughout the process continuous evaluation is performed to determine how to make the unit able to withstand the increasing stress. Generally temporary fixes are implemented just so that the test can continue. When a group of fixes is identified, a permanent block change is then implemented.

After one stimuli has been elevated to a level felt to be sufficient, another stimuli is selected for step stress testing. This progression continues until all stimuli have been applied separately. Then combined stresses are run to exploit the synergism between the stresses, that is, the combined effect may generate larger stresses than either stress alone would create. After design fixes for the identified problems have been implemented, a second series of step stresses are run to verify the fixes, assure that the fixes themselves have not introduced new problems and to look for additional problems which may have been missed due to the limited sample size. This aspect of HALT must be taken into account in selecting the appropriate stress levels since a slight increase in stress can greatly reduce the number of cycles to failure.

For all of these stimuli, the upper and lower operating limits and the destruct limits should be found or at least understood. Understood means that although the limits are not actually found, they are verified to be well beyond the limits which may be used in any future HASS test and even farther beyond the normal field environments. For example, a product may be able to withstand an hour of random vibration at 20 G_{rms} without failure. Although the destruct limit may not have been found, it is certainly high enough for most commercial equipment intended for non-military environments where the screen environment may be 10 G_{rms} random vibration for 5 minutes and the worst field environment is a truck ride while in an isolation container. This example of the capability far exceeding the field environment is quite common when HALT is properly applied.

There are several reasons for ascertaining both the operating limits and the destruct limits. Knowledge of the operating limits is necessary in order to assess if suitable design margins exist and how large the margins are likely to be as a function of population. It is also necessary to formulate failure detection tests. These can be run during any future HASS test since the

detection tests run during stimulation are necessary for high detectability of precipitated defects. Knowledge of the destruct limits is required in order to determine the design margins in non-operating environments and to assure that any future HASS environments are well below destruct levels.

8.7.6.4 HASS Testing

HASS is a form of accelerated environmental stress screening. It presents the most intense environment of any seen by the product, but it is typically of a very limited duration. HASS is designed to go to "the fundamental limits of the technology." This is defined as the stress level at which a small increase in stress causes a large increase in the number of failures. An example of such a fundamental limit might be the softening point of plastics.

HASS requires that the product have a sufficient margin of strength above that required to survive the normal use environments. Temperature, vibration levels, voltage and other stimuli exceeding the normal levels are used in HASS to force rapid defect precipitation in order to make the screens more effective and economical. The use of HASS requires a thorough knowledge of the product's ability to function at the extended ranges of simulation and also detailed knowledge about the failure mechanisms which limit these stimuli levels. Design and process changes are usually made to extend the functional and destruct levels of the equipment in order to assure large design and process margins as well as to allow HASS, with its attendant cost savings, to be performed. These saving can potentially produce orders of magnitude reduction in screening cost as well as significant quality improvements. One risk is that the item may be overdesigned.

Development of screening levels to be used in HASS begins during HALT testing. Operational levels and destruct levels are used as guidelines to select environmental limits during HASS. Two levels of environmental stimuli are chosen for each accelerated screening environment: the precipitation level and the detection level. Precipitation is the manifestation of a latent, or dormant, product flaw (i.e., it changes from a latent state to a patent or evident, detectable state). Detection is the observation that an abnormality exists. The observation may be made visually, electronically, audibly, etc.

The precipitation levels are chosen to be well below the destruct level, but beyond the operational limits. During the precipitation screen, the test item may not operate within the required limits but functional operation must be maintained and it must be monitored. These levels serve as the acceleration factor to minimize the time necessary to precipitate faults. The detection stress level is chosen outside of or just below the operational level determined during HALT testing. During the detection portion of the screen, operational parameters are monitored for compliance with the requirements. Once the screening parameters have been set, a proof-of-screen test must be performed to ensure that the accelerated screening levels are not damaging the product. The proof-of-screen is performed by simply running multiply accelerated screening profiles until either the product wears out or assurance is gained that the screen is not devouring appreciable useful life. Typically, repeating the screening environment 10 times is acceptable

proof, provided there is no evidence of product wear out.

Its is critical that the product be powered up and monitored during HASS. A large portion, typically greater than 50%, of the faults identified during screening are soft or intermittent faults. Not having complete diagnostics and detection these faults can be disastrous. An intermittent fault in the factory is very likely to be an early failure in the field.

HASS is a time compressed environmental stress screen applied at the earliest functional level of assembly. Complete functional monitoring of the test item is extremely important. Non-detected faults correlate with early life failures and dissatisfied customers. A poorly designed screen can be worse than no screen at all! Thus it is important to perform proof-of-screen evaluations prior to screening in production, to ensure that the screen does not appreciably reduce the useful life of the product. One must be receptive to changing the screen if field data indicates that a specific failure mechanism is escaping the screen. Thus an effective screening process is a dynamic process.

8.7.6.5 HAST (Highly Accelerated Temperature and Humidity Stress Test)

With the vast recent improvements in electronics technology and the speed with which these technology improvements are occurring, accelerated tests which were designed just a few years ago may no longer be adequate and efficient for today's technology. This is especially true for those accelerated tests intended specifically for microelectronics. For example, due to the improvements in plastic IC packages, the previous virtually universally accepted 85°C/85%RH Temperature/Humidity test now typically takes thousand of hours to detect any failures in new integrated circuits. In most cases the test samples finish the entire test without any failures. A test without any failures tells us very little. Yet we know that products still fail occasionally in the field; thus, we need to further improved our accelerated tests.

Without test sample failures we lack the knowledge necessary to make product improvements. Therefore the accelerated test conditions must be redesigned accordingly (e.g., utilize higher temperatures) to shorten the length of time required for the test, to make it more efficient and hence more cost effective. This is the background for today's focus (at the component level) upon Highly Accelerated Temperature and Humidity Stress Testing.

8.7.7 Accelerated Testing Data Analysis and Corrective Action Caveats

An accelerated test model is derived by testing the item of interest at a normal stress level and also at one or more accelerated stress levels. Extreme care must be taken when using accelerated environments to recognize and properly identify those failures which will occur in normal field use and conversely those that are not typical of normal use. Since an accelerated environment typically means applying a stress level well above the anticipated field stress, accelerated stress can induce false failure mechanisms that are not possible in actual field use. For example, raising the temperature of the test item to a point where the material properties change or where a

dormant activation threshold is exceeded could identify failures which cannot occur during normal field use. In this situation, fixing the failure may only add to the product cost without an associated increase in reliability. Understanding the true failure mechanism is paramount to elimination of the root cause of the failure.

The key to a successful accelerated testing program is to properly identify the failure mechanism and then eliminate the fault. Accelerating an environment such as temperature or vibration will uncover a multitude of faults. Each of these faults must be analyzed until the failure mechanism is fully understood. Chasing the wrong failure mechanism and implementing corrective action which does not eliminate the true cause of failure adds to the product's cost but does not improve product reliability.

A systematic method of tracking faults identified during accelerated testing ensures that problems are not forgotten or conveniently ignored. Each fault must then be tracked from the moment it is identified until either: a) corrective action is verified and documented or, b) a decision is made not to implement correction action. The failure tracking system must be designed to track the short term progress of failures over time.

When quantitative estimate of life or reliability is needed, the failure distribution must be determined for each stress condition. Next a model is derived to correlate the failure distributions. This is done to quantitatively predict performance under normal use, based upon the observed accelerated test data.

Constant stress prediction models frequently employ a least-square fit to the data using graphical methods such as those previously described in Section 8.3.1 or statistical methods such as those described in Section 8.3.2. However, when non-constant stresses are used, correctly plotting the data is much more complicated. Also, in many cases it may be necessary to use more elaborate techniques, such as those described in Section 8.3.2.4, to account for censored data.

Censored data is defined as data for test specimens which do not have a recorded time to failure. Some of the reasons for censoring data include:

(1) A unit may still be running without failure when the test ends

(2) The failure may be for some reason other than the applied test stress (e.g. mishandling)

(3) The item may have been removed from the test before failure for various reasons.

Complex censored data cases usually require powerful analysis tools, e.g., maximum likelihood methods, and cumulative damage models. Such tools can be cumbersome to use, but fortunately there are a number of statistically based computer programs to assist in these analyses.

Identifying which corrective action will solve the problem frequently involves multiple

engineering and production disciplines. Multiple discipline involvement is necessary to prevent finding a "fix" which cannot be economically built in production. Corrective action frequently involves a pilot build process which confirms that the "fix" does not introduce unanticipated new problems.

Corrective action verification should be performed in quick steps whenever possible. The accelerated testing environment is reapplied to verify that the proposed corrective action does eliminate the problem. Documenting the action taken is necessary to prevent reoccurrence and to ensure that production is modified to conform to the design change. Documentation should be shared throughout the organization to ensure that reoccurrence is indeed prevented. Conversely, a decision might be made not to implement corrective action based upon a monetary risk assessment.

Corrective action is expensive, if the problem affects only a small portion of the product population, the anticipated warranty repair cost will probably also be low. Thus the program management may elect to live with the identified risk. The decision, however, must always be based upon the root cause of the failure not applying to the intended use of the product, e.g., the failure mechanism cannot occur in normal field usage. This decision should always be made with due caution. Historically, some "non-relevant" or "beyond normal use" failures do recur in the field and become very relevant.

8.8 References for Section 8

1. Engineering Design Handbook: Reliability Measurement. January 1976, AMCP-706-198, AD#A027371.

2. Horn, R., and G. Shoup, "Determination and Use of Failure Patterns," Proceedings of the Eighth National Symposium on Reliability and Quality Control, January 1962.

3. VanAlvin, W. H., ed., Reliability Engineering. Englewood Cliffs, NJ: Prentice-Hall Inc., 1966.

4. Lloyd, R.K. and M. Lipow, Reliability: Management, Methods, and Mathematics, TRW, Redondo Beach, CA, second edition, 1977.

5. Mann, N., R. Schafer and N. Singpurwalla, Methods of Statistical Analysis of Reliability and Life Data. New York, NY: John Wiley and Sons, 1974.

6. Quality Assurance Reliability Handbook. AMCP 702-3, U.S. Army Materiel Command, Washington DC 20315, October, 1968, AD#702936.

7. Turkowsky, W., Nonelectronic Reliability Notebook. RADC-TR-69-458, March 1970.

8. Hobbs, G. K., Highly Accelerated Life Tests - HALT, unpublished, contained in seminar notes, "Screening Technology" © April 1990.

9. Harris, C.M., Crede, C.E, "Shock and Vibration Handbook," McGraw-Hill, 1961.

10. Nelson, Dr. Wayne, "Accelerated Testing," John Wiley & Sons, 1990.

11. "Sampling Procedures and Tables for Life and Reliability Testing Based on the Weibull Distribution (Mean Life Criterion)," Quality Control and Reliability Technical Report, TR3, Office of the Assistant Secretary of Defense (Installations and Logistics), September 30, 1961.

12. "Sampling Procedures and Tables for Life and Reliability Testing Based on the Weibull Distribution (Hazard Rate Criterion)," Quality Control and Reliability Technical Report TR4, Office of the Assistant Secretary of Defense (Installations and Logistics), February 28, 1962.

13. "Sampling Procedures and Tables for Life and Reliability Testing Based on the Weibull Distribution (Reliable Life Criterion)," Quality Control and Reliability Technical Report, TR6, Office of the Assistant Secretary of Defense (Installations and Logistics), February 15, 1963.

14. Crow, L. H., "On Tracking Reliability Growth," Proceedings 1975 Annual Reliability & Maintainability Symposium, pp 438-443.

15. Discrete Address Beacon System (DABS) Software System Reliability Modeling and Prediction, Report No. FAA-CT-81-60, prepared for U.S. Department of Transportation, FAA Technical Center, Atlantic City, New Jersey 08405, June 1981.

16. Reliability Growth Study, RADC-TR-75-253, October 1975, ADA023926.

17. Green, J. E., "Reliability Growth Modeling for Avionics," Proceedings AGARD Lecture Series No 81, Avionics Design for Reliability, April 1976.

18. MIL-HDBK-781A, "Reliability Test Methods, Plans and Environments for Engineering, Development, Qualification and Production," April 1996.

THIS PAGE HAS BEEN LEFT BLANK INTENTIONALLY

9.0 SOFTWARE RELIABILITY

9.1 Introduction

Hardware reliability engineering was first introduced as a discipline during World War II to evaluate the probability of success of ballistic rockets. The 1950's brought more advanced methods to estimate life expectancies of mechanical, electrical and electronic components used in the defense and aerospace industry. By the 1960's, reliability engineering had established itself as an integral part of end user product development in commercial products as well as military applications. (Ref. [1]).

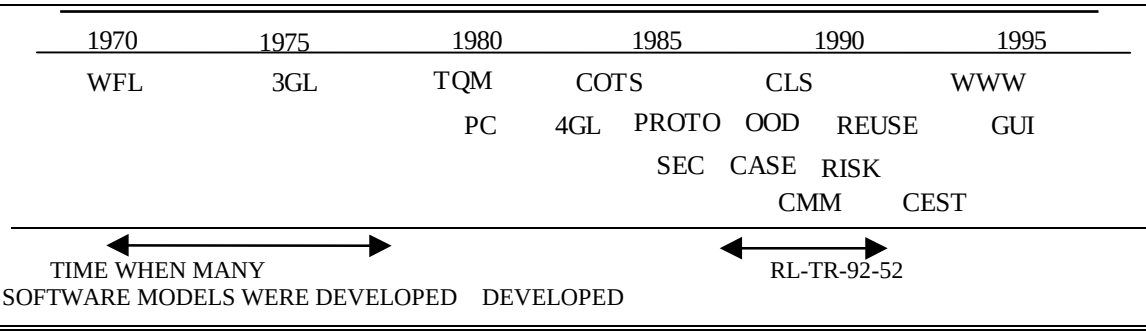
The *software reliability* discipline is much younger, beginning in the mid 1970's when the software development environment was reasonably stable. Most of software reliability models were developed during this time of software stability. However, a surge of new technology, new paradigms, new structured analysis concepts, and new ways of developing software emerged in the late 1980's and continues to this date. Figure 9.1-1 provides a chronological reference for some of the elements which comprise the current software development environment and add to its complexity.

As more and more systems that are a part of everyday life become more and more dependent upon software, perceptions about software reliability have changed. Increasing control by software of items such as dishwashers, ovens and automobiles, along with liability issues associated with these products, has led to an increased awareness of the criticality of reducing "hidden" software errors. Additionally, the influx of computers into financial and security-related operations requires a guarantee of data integrity.

Software engineers uniformly do not have an analogous view of reliability. Webster defines reliable as "giving the same result on successive trials." This definition, when extrapolated to include "forever," more closely resembles the view of reliability imposed on software engineers. In general, the reliability metric for software is used to describe the probability of the software operating in a given environment within the designed range of input without failure. Therefore, *software reliability* is defined as the probability that software will not cause a system failure over a specified time under specified conditions. This probability is a function of the inputs to and use of the system, as well as the presence of latent software faults. The system inputs determine whether any latent faults will be encountered during system operation.

SECTION 9: SOFTWARE RELIABILITY

WHEN THESE SOFTWARE ENGINEERING CONCEPTS WERE INTRODUCED



LABEL	TERM	LABEL	TERM
3GL	3rd Generation Languages	PC	Personal Computers
4GL	4th Generation Languages	PROTO	Rapid Prototyping
CASE	Computer Assisted Software Engineering Tools	REUSE	Reuse
CEST	Software Cost Estimation	RISK	Software Risk Management
CLS	Client Server Technology	RL-TR-	Software Reliability, Measurements and Testing
CMM	Capability Maturity Model	SEC	Software Engineering Curricula
COTS	Commercial Off-the-Shelf Software	TQM	Total Quality Management
GUI	Graphical User Interfaces	WFL	Waterfall Development Model
OOD	Object Oriented Design	WWW	World Wide Web (Internet)

FIGURE 9.1-1: SOFTWARE ENVIRONMENT TIMELINE

Additional differences between hardware and software reliability include:

(1) The age of the software has nothing to do with its failure rate. If the software has worked in the past, it will work in the future, everything else remaining the same (i.e., no hardware, software or interface changes). Software does not rust or exhibit other hardware wearout mechanisms.

(2) The frequency of software use does not influence software reliability. The same software can be used over and over and, if it did not fail the first time, it will not fail any other time in identical usage (same range of inputs with no hardware, software or interface changes). In contrast, physical parts wear from usage, resulting in failure.

(3) Software does become obsolete as user interface standards evolve and hardware become antiquated.

(4) With the exception of documentation and storage/transfer media, software, unlike hardware, cannot be held or touched. Typical methods of judging a hardware item include observing size and material composition, quality of assembly (form, fit and finish), and compliance with specification. For example, one can observe how well two gears mesh or if a transistor has sufficient current capacity for a circuit application. These physical concepts do not apply to software.

(5) Software cannot be judged prior to use by the same methods as hardware, i.e., there is no equivalent to incoming inspection.

(6) Software must be matched with hardware before it can ever be tested. If a failure occurs, the problem could be hardware, software, or some unintended interaction at the hardware/software interface.

(7) In general, hardware will either work or not in a given application. Software, aside from total failure, has varying degrees of success according to its complexity and functionality.

(8) Although not executable, documentation usually is considered an integral part of the software. Documentation which does not fully or accurately describe the operation can be considered to be just as much a failure as a software crash. When a user expects on-line help and does not get it (either because it is not activated or because what was provided was incorrect or incomplete), the software does not meet the user's expectation and, therefore, is not perfectly reliable. In contrast, documentation is usually not assessed when evaluating hardware reliability.

Admittedly there are differences between hardware and software. Rather than dwelling on the differences, we should look at the similarities. Some of these are:

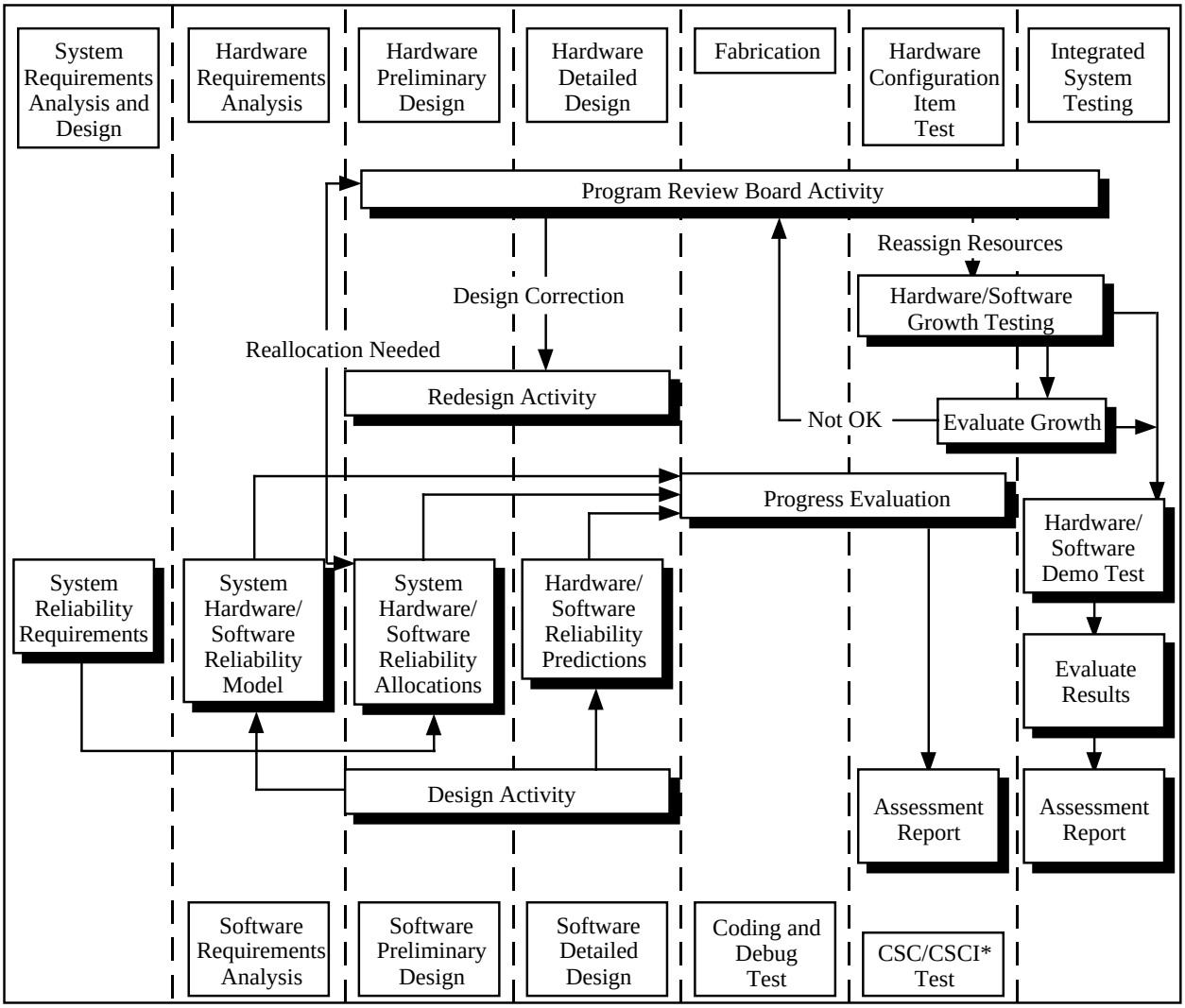
(1) Hardware reliability is a function of equipment complexity; intuitively one would expect the same to be true of software.

(2) Solid state electron devices (e.g., transistors, microcircuits) if fabricated properly, do not have any wearout mechanisms that one can see over a long time period. The defects which cause failure (other than obvious misapplication of the device) are built-in during the initial fabrication of the device; the same is true of software.

(3) Hardware reliability can be improved by reliability growth testing, e.g., a test-analyze-and-fix program to discover, identify, and correct failure modes and mechanisms which would cause early equipment failure. This is similar to finding and eliminating "bugs" in a software program, thus increasing its reliability.

Thus, we should be concentrating on the duality that exists between the successful hardware approaches and the emerging software approaches. Once this is accepted, the whole problem is simplified because the hardware and software problems can be approaches together in a total system context.

The duality between hardware and software is graphically portrayed in Figure 9.1-2 which illustrates the key elements of hardware and software programs during the life cycle phases of system development. The basic difference occurs during full scale engineering development, when hardware is fabricated and tested while software is coded (programmed) and debugged.



* Computer Software Component/Computer Software Configuration Item

FIGURE 9.1-2: HARDWARE/SOFTWARE SYSTEM LIFE CYCLE RELATIONSHIP (REF. [2])

9.2 Software Issues

Quality Focus. One essential concept for both hardware and software is that the customer's perception of quality is extremely important. Quality is delivering what the customer wants or expects. Customers must be considered during the specification and design stages of development. Since various customer groups have conflicting interests and view quality and reliability differently, it is important to analyze the customer base.

For example, the organization funding a project is one customer, the user another. If they are

different organizations, their expectations may be in conflict. Quality for the funding organization may be interpreted as "delivering on time and within budget" with "conformance to requirements" viewed as having less priority. In contrast, the customer who depends on the system's functionality to meet organizational needs is probably not as concerned with development schedule or cost. The pilot of a jet fighter expects the hardware and software to work perfectly regardless of whether the various sub-systems were delivered on time or within budget. Any failure, for any reason, may be catastrophic. On the other hand, those accountable for verifying that the jet will not fail are very much interested in ensuring that both the hardware and software have been thoroughly tested and that the reliability assessment process is consistent with what has been used in other systems that have proved to be as reliable as predicted. The expectation is that quality consists of evidence that everything possible has been done to ensure failure-free operation, providing very high reliability.

The Software Engineering Institute (SEI) Capability Maturity Model (CMM) provides a framework for organizing small evolutionary steps into five maturity levels. These levels provide successive foundations for continuous improvement. Details of each level are found in "Capability Maturity Model for Software (Version 1.1)," CMU/SEI-93-TR-024, Software Engineering Institute, and are summarized in the following paragraphs.

Level 1. At the initial level, Level 1, the organization typically lacks a stable environment for developing and maintaining software. In this case, the benefits of good software engineering practices are undermined by ineffective planning and reactive systems. Since the software process is not stable, the software process capability is unpredictable. Schedules, budgets, functionality, and product quality also are generally unpredictable.

Level 2. An organization at the repeatable level, Level 2, has developed policies for managing software projects and has procedures for implementing those policies. Experience gained on one software development project is used to plan and manage new, similar projects. One criteria for Level 2 is the institutionalization of effective management processes for software development. This institutionalization allows successful practices developed on earlier projects to be repeated, although specific processes may differ from project to project. An effective process has the following characteristics: practiced, documented, enforced, measured and improvable.

A Level 2 organization has basic software management controls in place. Managers of software projects track costs, schedule, and functionality. They monitor the project to identify problems in meeting commitments. Software requirements and associated work products are baselined and the integrity of the configuration is controlled. Defined project standards are available and faithfully followed. A strong customer-supplier relationship is established with any subcontractors.

Level 3. Level 2 is called the defined level. At this level, the standard process for developing and maintaining software throughout the organization is documented. Software engineering and management processes are integrated into a coherent whole. Effective

software processes are exploited in the development of the organization's standard software process. Training is conducted across the organization to ensure managers and staff have the knowledge and skills needed to carry out their role in the process. One group is responsible for the organization's software process activities.

The characteristics of a well-defined software process include readiness criteria, inputs, work performance standards and procedures, verification mechanisms, outputs, and completion criteria. A well-defined software process gives management good insight into technical progress.

Level 4. At the managed level, Level 4, quantitative defect goals for software and the software process are established. Productivity and defect rates for important software process activities are measured across all projects as part of an organization-wide measurement program. All measurement data is entered into a common data base and used to analyze process performance. Project managers control assigned projects and processes by reducing variations in performance to fall within acceptable limits. Risks associated with moving up the learning curve of a new application domain are known, tracked, and managed.

Level 5. The highest level of maturity is aptly called the optimizing level. Here the organization has the means and will to continuously improve the process. Weaknesses are identified and processes are strengthened proactively, with the prevention of defects being the objective. Data on the effectiveness of the software process are collected and used to conduct cost-benefit analyses of new technologies and proposed process changes. Innovative ideas that capitalize on the best software engineering practices are identified and implemented throughout the organization.

At Level 5, the software process capability is characterized as continuously improving. This continuous improvement results from constantly striving to improve the range of process capability, thereby improving process performance of projects. Improvement comes in the form of incremental advancement of existing processes and innovative application of new technologies and methods.

Organizational Structure. The typical sequential organizational structure does not support significant cross communication between hardware and software specialists. An organization's internal communication gap can be assessed by considering the questions in Table 9.2-1. The answers help determine if the organizational structure creates two "separate worlds." If reliability is important and a communication gap exists, then the organization needs to break down the communication barriers and get all parts of the technical community to focus on a common purpose. Activities may involve awareness training, cross training, organizational restructuring, implementing/improving a metrics program, reengineering the overall system development processes as well as the sub-system (i.e., hardware and software) processes, or instituting a risk assessment/risk management program.

Reliability Terminology. While hardware-focused reliability engineers have adopted a common set of concepts and terms with explicit meaning, the software community has not yet reached consensus and, hence, no universally adopted terminology set is in place. Many concepts, fundamental to the discussion and development of software reliability and quality, have several meanings. Worse, they are often used interchangeably!

TABLE 9.2-1: ASSESSING THE ORGANIZATIONAL COMMUNICATIONS GAP

- Is the software group a separate entity?
- Does the organization consider software as an engineering discipline?
- What is the career path for hardware/software, or system engineers?
- What forums exist for interaction engineers, and project managers?
- Who heads up system development? Hardware engineers? Software engineers? Others?
- Is there an expressed need for quantifying system reliability?
- Who has defined the system reliability metric?
- Who is responsible for assessing system reliability?
- What metrics are in place for assessing system reliability?
- What program is in place for testing system reliability?

For instance, software engineers often use "*defect*", "*error*", "*bug*", "*fault*", and "*failure*" interchangeably. Capers Jones (Ref. [3]) defined these terms as follows:

(1) Error: A mistake made by a programmer or software team member that caused some problem to occur.

(2) Bug: An error or defect that finds its way into programs or systems.

(3) Defect: A bug or problem which could cause a program to either fail or to produce incorrect results.

(4) Fault: One of the many nearly synonymous words for a bug or software defect. It is often defined as the manifestation of an error.

Some software specialists define a "*failure*" as any inappropriate operation of the software program while others separate "*faults*" and "*failures*" on a time dimension relative to when a defect is detected: "*faults*" are detected before software delivery while "*failures*" are detected after delivery. To the hardware community this appears to be an artificial distinction; yet it is important to be aware of the differentiation since both terms are used in actual practice. Software people talk about "*fault rate*" and "*failure rate*", with the latter term having a different meaning than that used with regard to hardware.

Robert Dunn (Ref. [4]) defines a software defect as "Either a fault or discrepancy between code and documentation that compromises testing or produces adverse effects in installation, modification, maintenance, or testing". In contrast, Putnam and Myers (Ref. [5]) define a defect as "A software fault that causes a deviation from the required output by more than a specified tolerance. Moreover, the software need produce correct outputs only for inputs within the limits that have been specified. It needs to produce correct outputs only within a specified exposure period." Since these definitions differ, a count of the number of defects will yield different results, and, hence, a different defect rate, depending on the counter's definition.

Dunn separates defects into three classes (he feels that it is fairly easy for experienced programmers to relate to each of these):

(1) Requirements Defects: Failure of software requirements to specify the environment in which the software will be used, or requirements documentation that does not reflect the design of the system in which the software will be employed.

(2) Design Defects: Failure of designs to satisfy requirements, or failure of design documentation to correctly describe the design.

(3) Code Defects: Failure of code to conform to software designs.

Typical requirements defects include indifference to the initial system state, incomplete system error analysis and allocation, missing functions, and unquantified throughput rates or necessary response times. The many kinds of design defects include misinterpretation of requirements specifications, inadequate memory and execution time reserves, incorrect analysis of computational error, and infinite loops. Possible code defects include unreachable statements, undefined variables, inconsistency with design, and mismatched procedure parameters.

Other software experts have different classifications. For example, Putnam and Myers define six classes of defects:

(1) Requirements Defects
(2) Design Defects
(3) Algorithmic Defects
(4) Interface Defects
(5) Performance Defects
(6) Documentation Defects

Life Cycle Considerations. Hardware reliability often assumes that the *hazard rate* (i.e., failure rate per unit time, often shortened to the failure rate) follows the "bathtub" curve, illustrated in Figure 9.2-1. Failures occur throughout the item's life cycle; the hazard rate initially is decreasing, then is uniform, and finally is increasing.

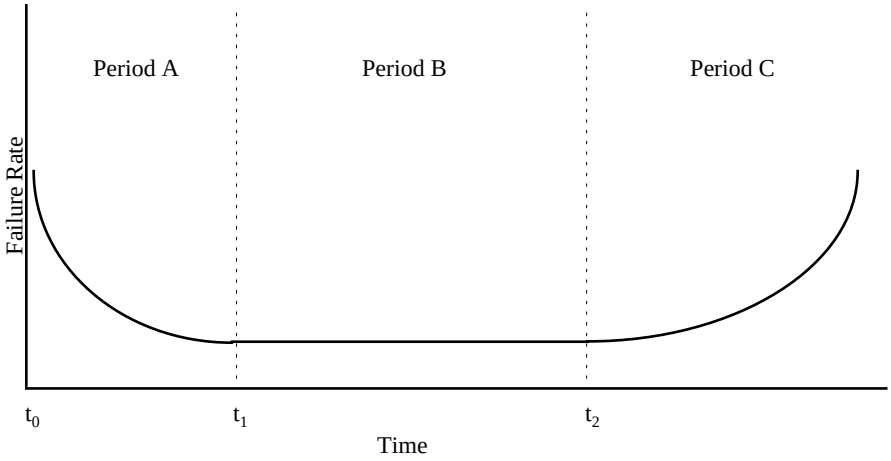


FIGURE 9.2-1: BATHTUB CURVE FOR HARDWARE RELIABILITY

The time points on the plot are defined as follows:

(1) Time t_0 is the time the population of components is activated or put into service ("fielded" or "distributed"); usually this is after the completion of development and production (whose times are not shown on the figure; i.e., design, build and test times are not included). Failures occurring during Period A, from t_0 to t_1, are said to be due to *infant mortality.*

(2) Time t_1 is the time when nearly all items with manufacturing defects have failed and have been removed from the population. Failures occurring during Period B, from t_1 to t_2, are assumed to be *random*, i.e., not due to any specific factor. The user is confident that the component will remain in service during this period. The probability that the component will function until time t_2 is expressed as the probability of success or the *reliability.*

(3) Time t_2 is the end of the *useful life* when components begin to exhibit end-of-life failures. Those failures occurring during Period C, after t_2, are considered to be due to *wearout*.

In hardware, the number of infant mortality failures observed in the field can be reduced by testing (*screening*) the components or assemblies prior to distribution (i.e., in the bathtub curve, the height of the curve in Period A can be reduced; alternatively the length of time attributable to infant mortality (Period A) can be reduced, causing t_1 to be moved closer to t_0). In the case of electronic components, this screen consists of operating, or *burning in*, the component for a time usually less than or equal to t_1. In the case of mechanical components, the screen may also include visual inspection. In addition, a random sample of the items may be tested to demonstrate adherence to specification. These procedures may be performed by the item

manufacturer prior to distribution to ensure that shipped components have few or no latent failures. Otherwise, the purchasing organization takes the responsibility for these activities.

When modeling the failure characteristics of a hardware item, the factors which contribute to the random failures must be investigated. The majority are due to two main sources:

(1) *Operating stress* is the level of stress applied to the item. The operating stress ratio is the level of stress applied relative to its rated specification. For example, a resistor rated to dissipate 0.5 watts when actually dissipating 0.4 watts is stressed at 80% of rated. Operating stresses are well defined and measurable.

(2) *Environmental stresses* are considered to be those due to the specific environment (temperature, humidity, vibration, etc.) that physically affect the operation of the item being observed. For example, an integrated circuit having a rated temperature range of 0° to 70°C that is being operated at 50°C is within operational environment specification. Environmental stresses also can be well defined and measurable.

When transient stresses occur in hardware, either in the operating stresses or the environmental stresses, failures may be induced which are observed to be random failures. For this reason, when observing failures and formulating modeling parameters, care must be taken to ensure accurate monitoring of all of the known stresses.

The same "*bathtub*" curve for hardware reliability strictly does not apply to software since software does not typically wearout. However, if the hardware life cycle is likened to the software development through deployment cycle, the curve can be analogous for times up to t_2. For software, the time points are defined as follows:

(1) Time t_0 is the time when testing begins. Period A, from t_0 to t_1, is considered to be the *debug* phase. Coding errors (more specifically, errors found and corrected) or operation not in compliance with the requirements specification are identified and resolved. This is one key difference between hardware and software reliability. The "clock" is different. Development/test time is NOT included in the hardware reliability calculation but is included for software.

(2) Time t_1 is the initial *deployment* (distribution) time. Failures occurring during Period B, from t_1 to t_2, are found either by users or through post deployment testing. For these errors, work-arounds or subsequent releases typically are issued (but not necessarily in direct correspondence to each error reported).

(3) Time t_2 is the time when the software reaches the end of its useful life. Most errors reported during Period C, after t_2, reflect the inability of the software to meet the changing needs of the customer. In this frame of reference, although the software is still functioning to its original specification and is not considered to have failed, that

specification is no longer adequate to meet current needs. The software has reached the end of its useful life, much like the wearout of a hardware item. Failures reported during Period C may be the basis for generating the requirements for a new system.

Usually hardware upgrades occur during Period A, when initial failures often identify required changes. Software upgrades, on the other hand, occur in both Periods A and B. Thus, the Period B line is not really "flat" for software but contains many mini-cycles of Periods A and B: an upgrade occurs, most of the errors introduced during the upgrade are detected and removed, another upgrade occurs, etc. Hence, Figure 9.2-2 might be a better representation of the software life cycle. Although the failure rate drops after each upgrade in Period B, it may not reach the initial level achieved at initial deployment, t_1. Since each upgrade represents a mini development cycle, modifications may introduce new defects in other parts of the software unrelated to the modification itself. Often an upgrade focuses on new requirements; its testing may not typically encompass the entire system. Additionally, the implementation of new requirements may inversely impact (or be in conflict with) the original design. The more upgrades that occur, the greater the likelihood that the overall system design will be compromised, increasing the potential for increased failure rate, and hence lower reliability. This scenario is now occurring in many legacy systems which have recently entered Period C, triggering current reengineering efforts.

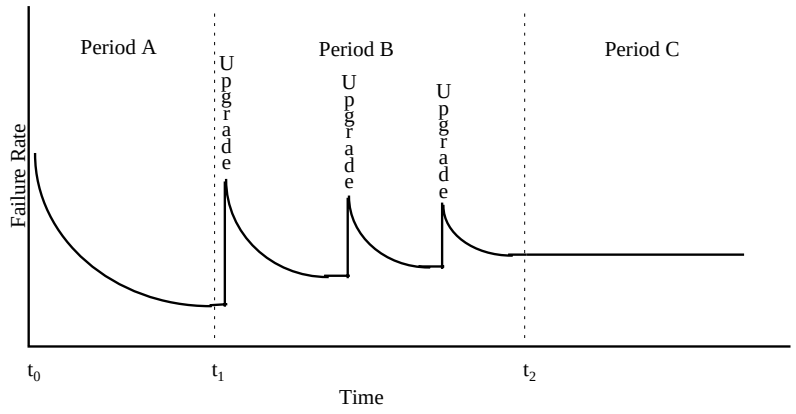


FIGURE 9.2-2: REVISED BATHTUB CURVE FOR SOFTWARE RELIABILITY

In software, the screening concept is not applicable since all copies of the software are identical. Additionally, typically neither operating stresses nor operational environment stresses affect software reliability. The software program steps through the code without regard for these factors. Other quality characteristics, such as speed of execution, may be effected, however. The end user might consider a "slow" program as not meeting requirements.

Table 9.2-2 summarizes the fundamental differences between hardware and software life cycles.

TABLE 9.2-2: SUMMARY: LIFE CYCLE DIFFERENCES

Life Cycle	Pre t_0	Period A (t_0 to t_1)	Period B (t_1 to t_2)	Period C (Post t_2)
HARDWARE	Concept Definition Development Build Test	Deployment Infant Mortality Upgrade	Useful Life	Wearout
SOFTWARE	Concept Definition Development Build	Test Debug/Upgrade	Deployment Useful Life Debug/Upgrade	Obsolescence

9.3 Software Design

Once the requirements have been detailed and accepted, the design will be established through a process of allocating and arranging the functions of the system so that the aggregate meets all customer needs. Since several different designs may meet the requirements, alternatives must be assessed based on technical risks, costs, schedule, and other considerations. A design developed before there is a clear and concise analysis of the system's objectives can result in a product that does not satisfy the requirements of its customers and users. In addition, an inferior design can make it very difficult for those who must later code, test, or maintain the software. During the course of a software development effort, analysts may offer and explore many possible design alternatives before choosing the best design.

Frequently, the design of a software system is developed as a gradual progression from a high-level or logical system design to a very specific modular or physical design. Many development teams, however, choose to distinguish separate design stages with specific deliverables and reviews upon completion of each stage. Two common review stages are the preliminary design and the detailed design.

9.3.1 Preliminary Design

Preliminary or high-level design is the phase of a software project in which the major software system alternatives, functions, and requirements are analyzed. From the alternatives, the software system architecture is chosen and all primary functions of the system are allocated to the computer hardware, to the software, or to the portions of the system that will continue to be accomplished manually.

During the preliminary design of a system, the following should be considered:

(1) Develop the architecture
- system architecture -- an overall view of system components
- hardware architecture -- the system's hardware components and their interrelations
- software architecture -- the system's software components and their interrelations

(2) Investigate and analyze the physical alternatives for the system and choose solutions

(3) Define the external characteristics of the system

(4) Refine the internal structure of the system by decomposing the high-level software architecture

(5) Develop a logical view or model of the system's data

9.3.1.1 Develop the Architecture

The architecture of a system describes its parts and the ways they interrelate. Like blueprints for a building, there may be various software architectural descriptions, each detailing a different aspect. Each architecture document usually includes a graphic and narrative about the aspect it is describing.

The software architecture for a system describes the internal structure of the software system. It breaks high-level functions into subfunctions and processes and establishes relationships and interconnections among them. It also identifies controlling modules, the scope of control, hierarchies, and the precedence of some processes over others. Areas of concern that are often highlighted during the establishment of the software architecture include: system security, system administration, maintenance, and future extensions for the system.

Another aspect of the software architecture may be the allocation of resource budgets for CPU cycles, memory, I/O, and file size. This activity often leads to the identification of constraints on the design solution such as the number of customer transactions that can be handled within a given period, the amount of inter-machine communication that can occur, or the amount of data that must be stored.

The first software architecture model for a system is usually presented at a very high level with only primary system functions represented. An example of a high-level software architecture is presented in Figure 9.3-1. As design progresses through detailed design, the architecture is continually refined.

9.3.1.2 Physical Solutions

Unless a software system has been given a pre-defined physical solution, an activity called environmental selection occurs during the preliminary design of a system. This is the process of investigating and analyzing various technological alternatives to the system and choosing a solution based upon the system's requirements, the users' needs, and the results of the feasibility studies. Aspects of a system that are generally selected at this time are: the hardware processing unit; computer storage devices; the operating system; user terminals, scanners, printers and other input and output devices; and the computer programming language.

In some cases, hardware and software items such as communications hardware and software, report writers, screen management systems, or database management systems are available "off-the-shelf." In other cases, unique requirements of the system may dictate the development of specific hardware and software items, specially designed for the system. The additional resources required to customize the system must be estimated and reviewed.

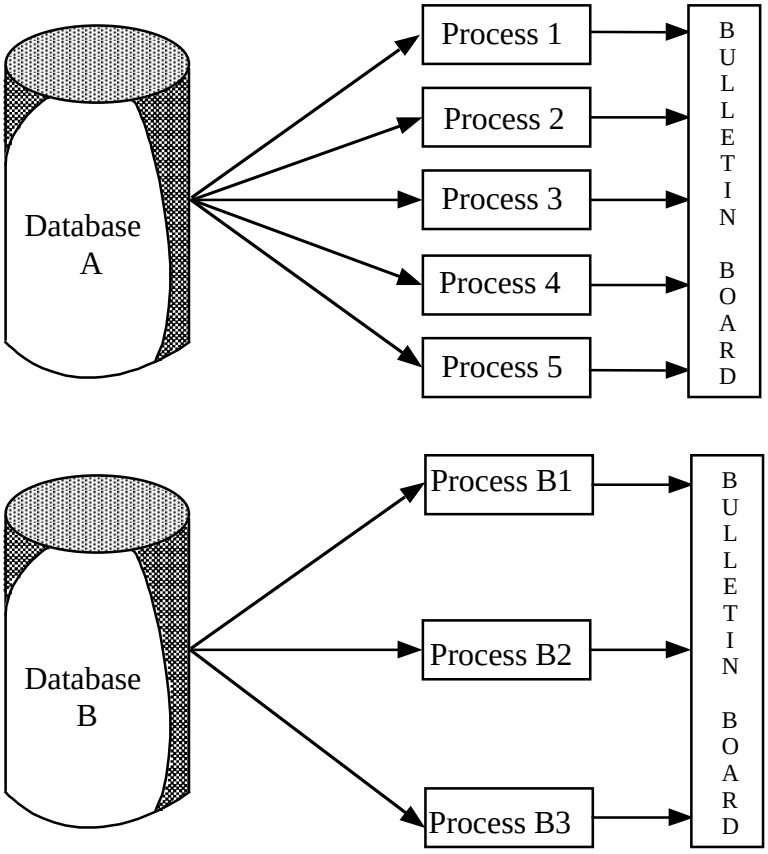


FIGURE 9.3-1: HIGH-LEVEL SOFTWARE ARCHITECTURE EXAMPLE

9.3.1.3 External Characteristics

Following the software system's functional allocation and physical environment selection, the details of the external or observable characteristics of a system can be developed. Included here would be terminal screen displays, report formats, error message formats, and interfaces to other systems.

A human factors engineer may be part of the design team concerned with the observable characteristics of a software system. This person specializes in the analysis of the human-machine interface. When a system's targeted users are novice computer users or when a system requires extensive manual data entry, human factors engineering can be a very important aspect of the design.

9.3.1.4 System Functional Decomposition

The activity of breaking a high-level system architecture into distinct functional modules or entities is called functional decomposition. When preparing to decompose a software system, the design team must decide what strategy they will use. Many decomposition strategies have been written about and are advocated; most the variations of the widely used top-down or bottom-up approaches. (Ref. [13]).

Top-down design is the process of moving from a global functional view of a system to a more specific view. Stepwise refinement is one technique used in top-down design. With this method, design begins with the statement of a few specific functions that together solve the entire problem. Successive steps for refining the problem are used, each adding more detail to the functions until the system has been completely decomposed.

A bottom-up design strategy for a software system is often used when system performance is critical. In this method, the design team starts by identifying and optimizing the most fundamental or primitive parts of the system, and then combining those portions into the more global functions. (Ref. [14] and [15]).

9.3.2 Detailed Design

Detailed design or low-level design determines the specific steps required for each component or process of a software system. Responsibility for detailed design may belong to either the system designers (as a continuation of preliminary design activities) or to the system programmers.

Information needed to begin detailed design includes: the software system requirements, the system models, the data models, and previously determined functional decompositions. The specific design details developed during the detailed design period are divided into three categories: for the system as a whole (system specifics), for individual processes within the system (process specifics), and for the data within the system (data specifics). Examples of the type of detailed design specifics that are developed for each of these categories are given below.

9.3.2.1 Design Examples

System specifics:

(1) Physical file system structure
(2) Interconnection records or protocols between software and hardware components
(3) Packaging of units as functions, modules or subroutines
(4) Interconnections among software functions and processes
(5) Control processing
(6) Memory addressing and allocation
(7) Structure of compilation units and load modules

Process specifics:

(1) Required algorithmic details
(2) Procedural process logic
(3) Function and subroutine calls
(4) Error and exception handling logic

Data specifics:

(1) Global data handling and access
(2) Physical database structure
(3) Internal record layouts
(4) Data translation tables
(5) Data edit rules
(6) Data storage needs

9.3.2.2 Detailed Design Tools

Various tools such as flowcharts, decision tables, and decision trees are common in detailed software design. Frequently, a structured English notation for the logic flow of the system's components is also used. Both formal and informal notations are often lumped under the term pseudocode. This is a tool generally used for the detailed design of individual software components. The terminology used in pseudocode is a mix of English and a formal programming language. Pseudocode usually has constructs such as "IF ..., THEN ...," or "DO ... UNTIL ...," which can often be directly translated into the actual code for that component. When using pseudocode, more attention is paid to the logic of the procedures than to the syntax of the notation. When pseudocode is later translated into a programming language, the syntactical representation becomes critical.

9.3.2.3 Software Design and Coding Techniques

Specific design and code techniques are related to error confinement, error detection, error recovery and design diversity. A summary of the each technique is included in Table 9.3-1 and Table 9.3-2.

TABLE 9.3-1: SOFTWARE DESIGN TECHNIQUES

Design Techniques
• Recovery designed for hardware failures
• Recovery designed for I/O failures
• Recovery designed for communication failures
• Design for alternate routing of messages
• Design for data integrity after an anomaly
• Design for replication of critical data
• Design for recovery from computational failures
• Design to ensure that all required data is available
• Design all error recovery to be consistent
• Design calling unit to resolve error conditions
• Design check on inputs for illegal combinations of data
• Design reporting mechanism for detected errors
• Design critical subscripts to be range tested before use
• Design inputs and outputs within required accuracy

TABLE 9.3-2: SOFTWARE CODING TECHNIQUES

Coding Techniques	
• All data references documented	• Temporary storage restricted to only one unit - not global
• Allocate all system functions to a CSCI	• Unit has single processing objective
• Algorithms and paths described for all functions	• Unit is independent of source of input or destination of output
• Calling sequences between units are standardized	• Unit is independent of prior processing
• External I/O protocol formats standardized	• Unit has only one entrance and exit
• Each unit has a unique name	• Flow of control in a unit is from top to bottom
• Data and variable names are standardized	• Loops have natural exits
• Use of global variables is standardized	• Compounded booleans avoided
• All processes within a unit are complete and self contained	• Unit is within standard on maximum depth of nesting
• All inputs and outputs to each unit are clearly defined	• Unconditional branches avoided
• All arguments in a parameter list are used	• Global data avoided
• Size of unit in SLOC is within standard	• Unit outputs range tested
• McCabe's complexity of units is within standard	• Unit inputs range tested
• Data is passed through calling parameters	• Unit paths tested
• Control returned to calling unit when execution is complete	

9.4 Software Design and Development Process Model

Software development can occur with no formal process or structure (called "ad hoc" development) or it can follow one of several approaches (i.e., methods or models). Ad hoc development usually is the default used by relatively inexperienced developers or by those who only develop software as an aside or on rare occasions. As developers become more experienced, they tend to migrate from operating in an ad hoc fashion to using more formal structured methodologies. These major software development process models have evolved based upon actual practice. The selection is based upon several basic concepts, as summarized in

Table 9.4-1 and described throughout this section.

However, it is important to realize that what is actually being practiced may not fully correspond to the theory of any one model. In reality, developers often customize a model by implementing one or a combination of several elements of the models described. What is important is to understand enough about what constitutes the organization's software development process to be able to identify what characterizes the process used and to determine whether it is stable. The process that is in place will determine not only what data are available but also when they are available and whether they are adequate for determining the software reliability and quality performance levels as defined by the customer's contract requirements.

TABLE 9.4-1: SOFTWARE DEVELOPMENT PROCESS SELECTION

Approach	When to Use
Waterfall Model or Classic Development Model	When the detailed requirements are known, and are very stable When the type of application has been developed before When the type of software class (e.g., compilers or operating systems) has been demonstrated to be appropriate When the project has a low risk in such areas as getting the wrong interface or not meeting stringent performance requirements When the project has a high risk in budget and schedule predictability and control
Prototyping Approach	When the input, processing, or output requirements have not been identified To test concept of design or operation To test design alternatives and strategies To define the form of the man-machine interface
Spiral Model	To identify areas of uncertainty that are sources of project risk To resolve risk factors To combine the best features of the classic model and prototyping
Incremental Model	When a nucleus of functionality forms the basis for the entire system When it is important to stabilize staffing over the life of the project
Cleanroom Model	When a project can be developed in increments When staff size is sufficient to perform independent testing (staff > 6) When the approach has management support

9.4.1 Ad Hoc Software Development

The reality in many organizations where software development is not the main focus is that the development process is *ad hoc*. This is a polite way of saying that a defined structured process does not exist. The development effort is subject to the habits and operating styles of the individuals who comprise the project team. Responsibility for the project, and for interaction with the customer, is often in the hands of a non-software engineer. The software is viewed as having a supporting role to the project as a whole. Communication regarding requirements is primarily verbal and seldom documented. It is assumed that requirements are understood by all parties. Additionally, requirements change throughout the development effort. There is seldom a focus on design; design and code become merged into one task. Testing is the responsibility of the development team, and is often reduced to a random selection of functionality because there is no time to do a thorough job. Documentation, including design documents, is often written after the code is completed, and then reflects what was developed rather than serving as a guide for development. The project schedule is often determined by who is available to work rather than who is best qualified, the amount of dollars available, and an arbitrary completion date that typically is derived from something other than the functionality to be developed. The driving force is "having something to show by a specified date."

9.4.2 Waterfall Model

The *Waterfall Model* is presented in Figure 9.4-1. In its most simplistic interpretation it suggests that the process is strictly sequential, that there is a flow of ideas through the phases, with each phase having a distinct beginning and end and each phase enhancing the development to result in a software product that is operational when the bottom of the waterfall is reached.

The original intention of this model was that the development process is stable if all rework requires going back only one step in the process in order to be rectified. For example, if analysis revealed that initial requirements were incomplete then further requirements gathering would be implemented. If a particular design could not be coded correctly in the given environment then the design would be revisited. Testing would uncover coding errors which would be fixed before final delivery. The model suggests that the phases follow a time line, but this does not allow for revisiting previous phases when a problem is discovered.

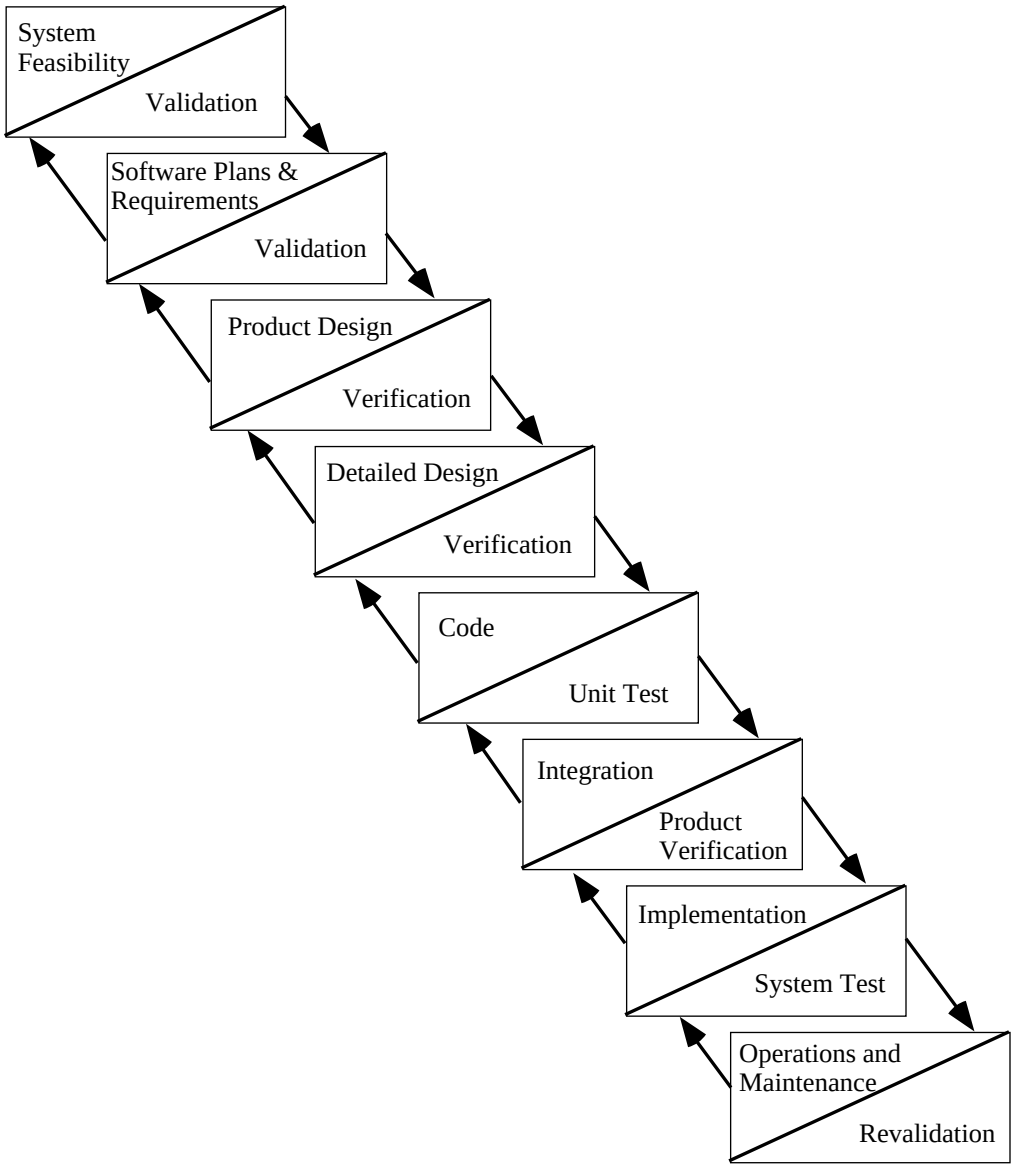
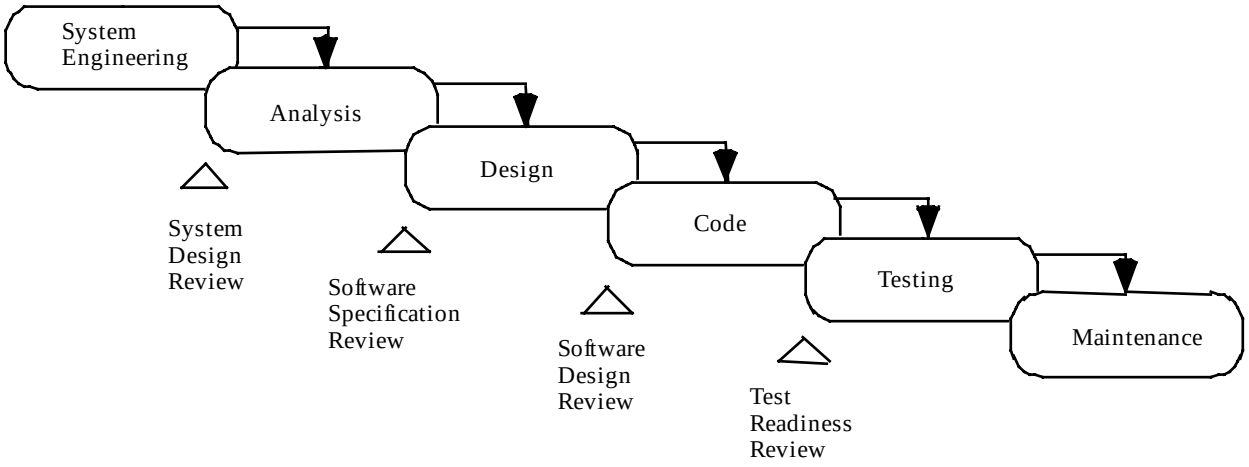


FIGURE 9.4-1: WATERFALL MODEL (REF. [6])

9.4.3 Classic Development Model

The *Waterfall Model* was later augmented to include precise phase ends and continuing activities, and has come to be known as the *Classic Development Model*; see Figure 9.4-2. This model provides a systemic approach to software development consisting of consecutive phases that begin with system engineering (sometimes called system requirements definition) and progress through requirements analysis, design, coding, testing, and maintenance. Each phase is defined in Figure 9.4-2.



PHASE	DESCRIPTION
System Engineering (sometimes called Requirements Definition)	When software is part of a larger system, work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This is essential since software must interface with other elements such as hardware, people, and databases. Requirements are defined at the system level with a small amount of top-level design and analysis. It is during this phase that developers identify previously developed subsystems that can be reused on the current system.
Requirements Analysis	The requirements definition process is now intensified and focused specifically on the software. The development team performs functional or object-oriented analysis and resolves ambiguities, discrepancies, and to-be-determined (TBD) specifications. To understand the nature of the software to be built, the developers must understand the information domains for the software, as well as the required functions, performance, and interfaces. Requirements for both the system and the software are documented and reviewed with the sponsor/user.
Design	Software design is actually a multi-step process that focuses on four distinct attributes of the software: data structure, software architecture, procedural detail, and interface characterization. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. During this step, the developers perform structured, data driven, or object-oriented analysis. Like requirements, the design is documented and becomes part of the software configuration.
Code	The design is translated (coded) into a machine-readable form. If design has been performed in a detailed manner, coding can be accomplished mechanistically. The developers also reuse existing code (modules or objects), with or without modification, and integrate it into the evolving system.
Test	Once new code has been generated or reused code has been modified, software testing begins. The unit test process focuses on the logical internals of the software, ensuring that all statements have been tested. The integration and system testing process focuses on the functional externals, testing to uncover errors and to ensure that the defined input will produce actual results that agree with required results. During acceptance testing, a test team that is independent of the software development team examines the completed system to determine if the original requirements are met. After testing the software is delivered to the customer.
Maintenance	Software may undergo change (one possible exception is embedded software) after it is delivered for several reasons (i.e., errors have been encountered, it must be adapted to accommodate changes in its external environment (e.g., new operating system), and/or customer requires functional or performance enhancements). Software maintenance reapplies each of the preceding phases, but does so in the context of the existing software.

FIGURE 9.4-2: THE CLASSIC DEVELOPMENT MODEL (REF. [7])

The *Classic Development Model* includes the notion of *validation* and *verification* at each of the phases. Validation is defined as testing and evaluating the integrated system to ensure compliance with the functional performance and interface requirements. Verification is defined as determining whether or not the product of each phase of the software development process fulfills all the requirements resulting from the previous phase. The purpose of the *validation* associated with the analysis and design model phases is to determine if the right product is being built. In revalidation activity that occurs after the software functionality has been defined, the purpose is to determine if the right product is still being built. *Verification* activity, associated with product design is to determine if the product is being built right, including the right components and their inter-combinations. This Classic Model has a definite and important role in software engineering history. It provides a template into which methods for analysis, design, coding, testing, and maintenance can be placed. It remains *the most widely used procedural model* for software engineering.

The classic model does have weaknesses. Among the problems that are sometimes encountered when the classic development process model is applied are:

(1) It emphasizes fully elaborated documents as completion criteria for early requirements and design phases. This does not always work well for many classes of software, particularly interactive end-user applications. Also, in areas supported by fourth-generation languages (such as spreadsheet or small business applications), it is unnecessary to write elaborate specifications for one's application before implementing it.

(2) Often the customer cannot state all requirements explicitly. The classic model requires this and has difficulty accommodating the natural uncertainty that exists at the beginning of many projects.

(3) The customer must have patience. A working version of the program is not available until late in the project schedule. Errors in requirements, if undetected until the working program is reviewed, can be costly.

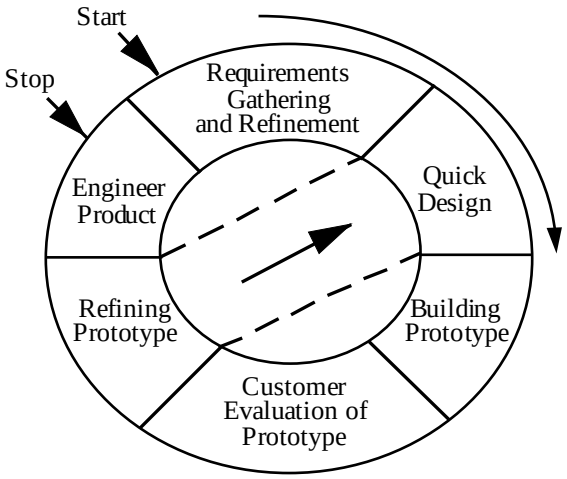
9.4.4 Prototyping Approach

Prototyping is a process that enables the developer to create a model of the software to be built. The steps for prototyping are identified and illustrated in Figure 9.4-3. The model can take one of three forms:

(1) A model that depicts the human-machine interaction in a form that enables the user to understand how such interaction will occur

(2) A working prototype that implements some subset of the functions required of the desired software

(3) An existing program that performs part or all of the functions desired, but has other features that will be improved upon in the new development effort



Step	Description
Requirements Gathering and Refinement	The developer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory.
Quick Design	The quick design focuses on a representation of those aspects of the software that will be visible to the user (e.g., user interface and output formats).
Prototype Construction	A prototype is constructed to contain enough capability for it to be used to establish or refine requirements, or to validate critical design concepts. If a working prototype is built, the developer should attempt to make use of existing software or apply tools (e.g., report generators, window manager) that enable working programs to be generated quickly.
Customer Evaluation	The prototype is evaluated by the customer and is used to refine requirements or validate concepts.
Prototype Refinement	The process of iteration occurs as the prototype is "tuned" to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

FIGURE 9.4-3: STEPS IN THE PROTOTYPING APPROACH

Using an iterative rapid prototyping approach, the concept of the software system gradually unfolds; each iteration continues to explore the functionality that is desired. This process is comparable to performing "what if" analyses. The developer uses the prototype to generate suggestions from users, including ideas for innovations and plans for revision of the prototype itself, or the process it supports.

Rapid prototyping can significantly improve the quality and reliability of software if the methodology is properly used. However there are severe adverse impacts to quality and reliability when the developer or the customer perceives the prototype to be the completed project. Adversaries of prototyping claim that prototyping should not replace the traditional development cycle for these reasons:

(1) If a system is needed badly the prototype may be accepted in its unfinished state and pressed into service without necessary refinement. Eventually, as deficiencies are realized, a backlash is likely to develop, requiring maintenance efforts which are extremely costly compared to the cost of doing it right the first time.

(2) It tends to shape the approach to a capability before it is thoroughly understood.

(3) The real costs of supporting prototypes after delivery are not well documented. Therefore there is little evidence to support statements claiming that the cost of software is less for systems developed under rapid prototyping methods.

(4) Prototyping can be difficult to manage as a project within a larger project.

The key to successful prototyping is to define the rules of the game at the beginning; that is, the customer and developer must both agree that the prototype is built to serve as a mechanism for defining requirements or validating critical design concepts. It is then discarded (at least in part) and the actual software is engineered with an eye toward quality and maintainability.

9.4.5 Spiral Model

The *Spiral Model* for software development is presented in Figure 9.4-4. This model has been developed to encompass the best features of both the Classic Model and prototyping, while at the same time adding the element of risk analysis that is missing in both these process models. In the simplest sense it represents the normalization of a “trial and error” methodology. It is used to explore the possibilities in situations where a need exists but the exact requirements are not yet known.

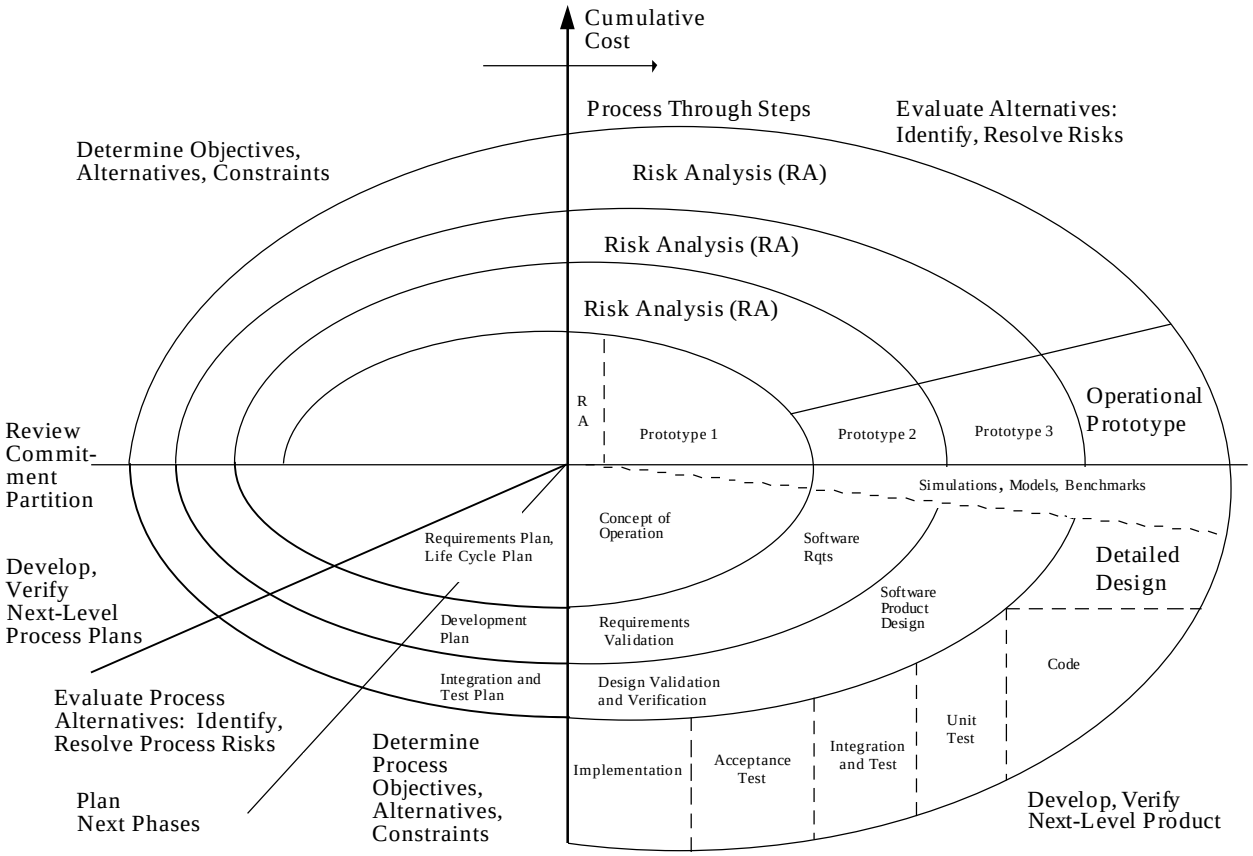


FIGURE 9.4-4: SPIRAL MODEL (REF. [7])

The model defines four major activities represented by the four quadrants (starting at upper left and progressing clockwise) of the figure:

(1) Planning: Determination of objectives, alternatives and constraints

(2) Risk analysis: Analysis of alternatives and identification/resolution of risks

(3) Engineering: Development of the "next-level" product

(4) Customer evaluation: Assessment of the results of engineering

With each iteration around the spiral (beginning at the center and working outward), progressively more complete versions of the software are built. During the first cycle around the spiral, objectives, alternatives, and constraints are defined and risks are identified and analyzed. If risk analysis indicates that there is uncertainty in requirements, prototyping may be used in the engineering quadrant to assist both the developer and the customer. Simulations and other models may be used to further define the problem and refine requirements. The customer evaluates the engineering work and makes suggestions for modification. Based on customer

input, the next phase of planning and risk analysis occur. At each cycle around the spiral, the culmination of risk analysis results in a "go, no-go" decision. If risk is too great, the project can be terminated. However, if the flow around the spiral path continues, each path moves the developer outward toward a more complete model of the system, and, ultimately, to the operational system itself. Every cycle around the spiral requires engineering that can be accomplished using either the classic or prototyping approaches.

Like the other development process models, the spiral model is not a panacea. The following are some of the reasons why it is not right for all developments:

(1) It may be difficult to convince the sponsor that the evolutionary approach is controllable.

(2) It demands risk assessment expertise, and relies on this expertise for success.

(3) If major risk areas are not uncovered during risk analysis, problems will undoubtedly occur.

(4) The model itself is relatively new and has not been used as widely as the Classic or prototyping approaches. It will take a number of years before its effectiveness and efficiency can be determined with certainty.

9.4.6 Incremental Development Model

The *Incremental Development Model* can be followed using a sequential approach or an iterative approach. In a sequential approach, once a step has been completed, a developer never returns to that step or to any step previous to that step. In an iterative approach, if there is sufficient reason to do so, the developer may return to a previously completed step, introduce a change, and then propagate the effects of that change forward in the development. Projects actually can rarely follow the sequential forward flow. Iteration is generally necessary.

The *Incremental Development Model* is based on developing the software in increments of functional capability with a series of overlapping developments and a series of staggered deliveries. As indicated in Figure 9.4-5, each increment is developed under the phased approach described for the Classic Development Model. Each increment undergoes structural, or top-level design, detailed design, code and unit test, integration and test, and delivery. The nucleus of the software, the "cornerstone" functionality that is the foundation for use, must be addressed in the structural design of the first increment. Additional capability is then added with successive increments. Note that all software efforts do not lend themselves to incremental development because it is often not possible to distinguish a nucleus of functional capability.

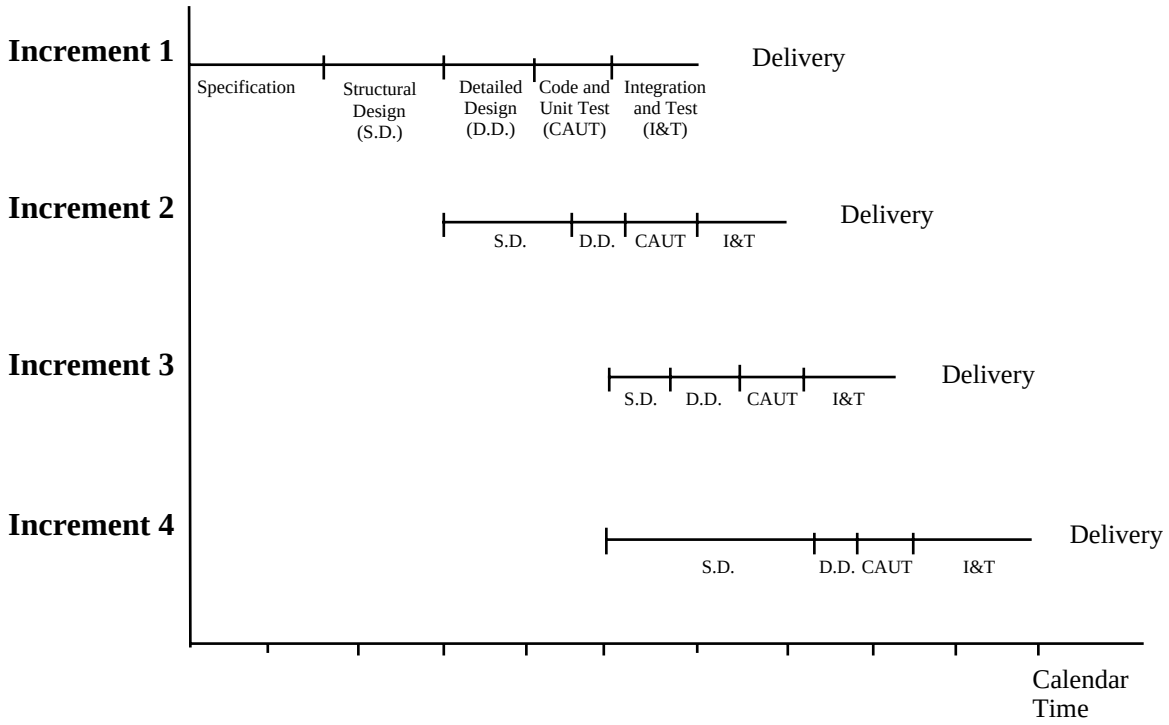


FIGURE 9.4-5: INCREMENTAL DEVELOPMENT MODEL (REF. [7])

Incremental development has been used successfully on many large projects. It is frequently used when the technical risks make it difficult to predict time scales for development, or when there is uncertainty about some aspects of the project. This approach also tends to level out or flatten the project's labor distribution curve. The design, program, and test teams can remain at relatively constant strength dealing with each increment in turn. Additionally, increments are easier to test and the cost of refinements is less expensive than with the single-shot Classic Development Model.

Incremental development is a useful approach when some functions within the software system have more stringent reliability requirements than others. Design efforts for a given increment can focus on attaining the desired reliability. Another feature of incremental development is that while the timeframe from project start to end may be identical to that of a project developed with the classic model, this model places operational software in the customer's hands long before project end.

9.4.7 Cleanroom Model

Cleanroom Software Engineering (Ref. [8] and [9]) (CSE) or just "*Cleanroom*" is a metaphor that comes from the integrated circuit manufacturing process where the environment must be free from all contaminants. If one were to rank all software development methodologies according to the amount of structure inherent in the methodology, the ad hoc development would be the lower bound (lack of structure) and cleanroom methodology would be the upper bound (very structured). Figure 9.4-6 illustrates the essential steps of the cleanroom development process.

The uniqueness of this approach is that it has embedded principles of total quality such as the use of teams, use of statistical process control techniques, and the commitment to "Do the right things right the first time" into the development process. The approach focuses on the aspects of the development that have the greatest impact on quality. Software reliability is specifically defined and measured as part of the certification process. Cleanroom Certification Test Teams provide scientific certification of software reliability -- they do not test it in.

Cleanroom methodology is premised on the notion that the best way to produce software approaching zero defects is to focus on defect prevention by clarifying requirements, developing precise functional and usage specifications, and then using them as the guide for planning and design, and for test development. It further presumes that correctness verification of the design will detect and eliminate most remaining significant defects before the software is actually built. The design effort entails writing pseudo code which is then subjected to correctness verification. The resulting pseudo code is so thorough and precise that it can be easily translated into the specified language. The actual coding is considered to be trivial relative to the development of pseudo code because the complex logic is addressed during pseudo code development.

In this methodology, the focus of testing reflects usage, not the structure of the software. Usage is inherent in the execution behavior of the software. Statistical Usage Testing is a process of testing software the way users intend to use it. The entire focus is on external system behavior, not on the internals of design and implementation. Test cases are randomly generated based on probability distributions that model anticipated software use in all possible circumstances including unusual and stressed situations. By definition, testing is designed to detect the more serious and/or high frequency defects first. Thus this testing method is more effective at improving software reliability in less time than traditional testing techniques. Data recorded includes execution time up to the point of each failure in appropriate units (measured in Central Processing Unit (CPU) time, clock time, or number of transactions, etc.). After execution of the test runs, the results are assessed and quality and performance measures computed.

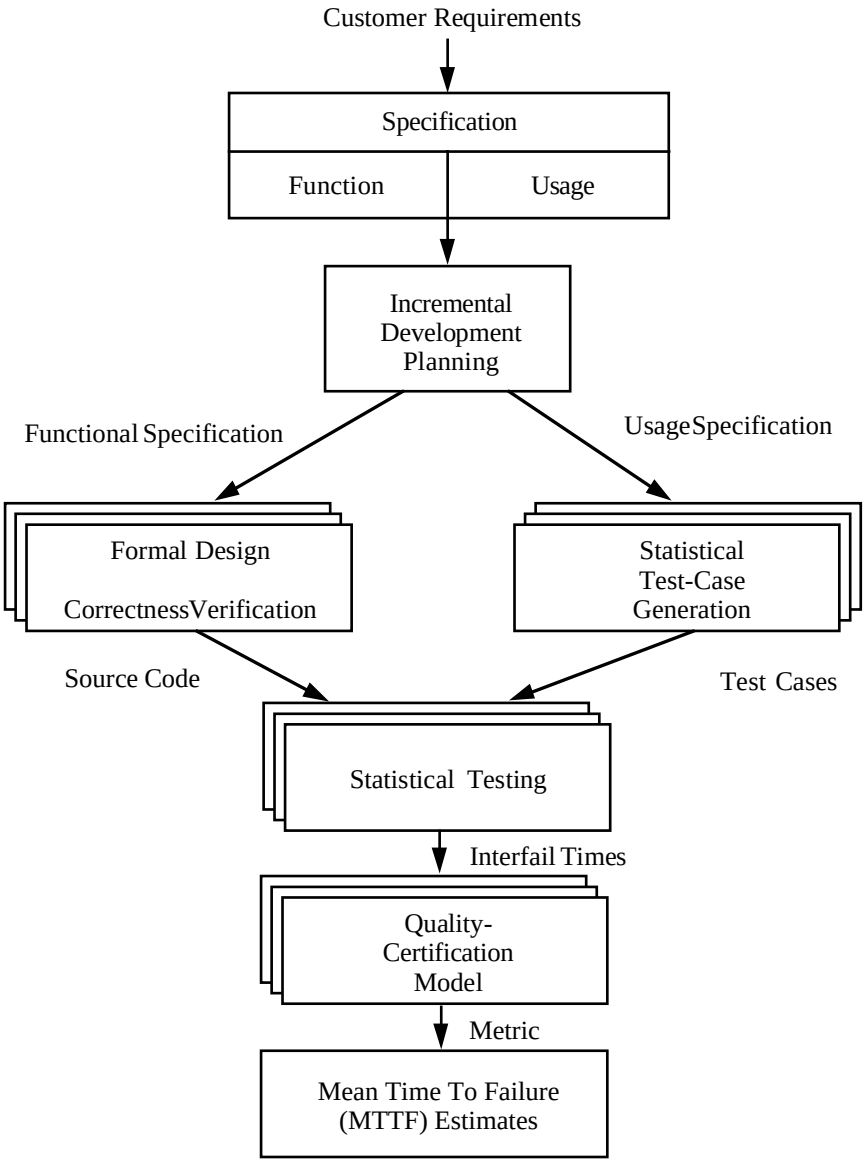


FIGURE 9.4-6: THE CLEANROOM DEVELOPMENT PROCESS (REF. [10])

Advocates for this methodology claim good results, namely that it is possible to produce software that approaches zero defects and deliver it on time and within budget. There is no way of knowing if a software system has zero defects, but as failure-free executions occur during testing, given the completeness of test coverage, one can conclude that there is a high probability that the software is at or near zero defects and will not fail during actual usage. The cleanroom approach has been adopted by more than 30 software development organizations as noted in the "Software Technology Support Center Guide (1995)." The cleanroom methodology has been applied to new systems, maintenance and evolution of existing systems, and re-engineering of problem systems. As of the end of 1993, cleanroom methodology used to develop a variety of projects totaling more than one million lines of code has shown extraordinary quality compared to

traditional results. A summary of cleanroom performance measures is given in Table 9.4-2. It should be noted, however, that these results are achieved with cleanroom teams composed of adequately trained journeyman programmers.

TABLE 9.4-2: CLEANROOM PERFORMANCE MEASURES (REF. [11])

Software Development Practices	**Defects During Development** (defects per KLOC*)	**Operational Failures** (failures per KLOC)	**Resultant Productivity** (LOC*/Staff Month)
Traditional Software-as-art	50 - 60	15 - 18	Unknown
Software Engineering	20 - 40	2 - 4	75 - 475
Cleanroom Engineering	0 - 5	< 1	> 750

*** KLOC - Thousand Lines of Code * LOC - Lines of Code**

Adversaries claim that it is an unrealistic methodology for the following reasons:

(1) The required statistical knowledge is beyond the realm of most software engineers.

(2) The testing strategies are too complicated to expect the average developer to use.

(3) It is too complicated for use on small projects.

(4) The paradigm shift required is so radical that software people will never accept it.

Software Reliability Prediction and Estimation Models

Software reliability models have been in existence since the early 1970's; over 200 have been developed. Certainly some of the more recent ones build upon the theory and principles of the older ones. Some of the older models have been discredited based upon more recent information about the assumptions and newer ones have replaced them. This review of software reliability is not meant to be an exhaustive review of every model ever developed but, rather, a discussion of some of the major models in use today, highlighting issues important to the reliability engineer.

Prediction vs. Estimation Models

Software reliability modeling is generally used for one of two purposes: to make *predictions* and for *estimation*. Software reliability prediction models use historical data for similar systems while estimation models use data collected during test. Prediction, therefore, is usually less accurate than estimation. The objective of software prediction is to predict the potential reliability (fault rate) early in the development process. Insight into potential reliability allows

improvements in software management to be considered before coding and testing start. The objective of the estimation process is to determine the number of faults remaining in the software just prior to testing so that the length of the test can be determined. Table 9.5-1 provides a comparison of prediction and estimation models.

TABLE 9.5-1: COMPARING PREDICTION AND ESTIMATION MODELS

Issues	Prediction Models	Estimation Models
Data Reference	Uses historical data	Uses data from the current software development effort
When Used In Development Cycle	Usually made prior to development or test phases; can be used as early as concept phase	Usually made later in life cycle (after some data have been collected); not typically used in concept or development phases
Time Frame	Predict reliability at some future time	Estimate reliability at either present or some future time

9.5.1 Prediction Models

The most basic prediction model involves the use of an organization's internal data, based on extensive experience and tracking, to develop predictions. Four other prediction models have been developed: *Musa's Execution Time Model*, (Ref. [12]), *Putnam's Model*, (Ref. [5]), and two models developed at Rome Laboratory and denoted by their technical report numbers: the *TR-92-52 Model* (Ref. [16]) and the *TR-92-15 Model* (Ref. [17]). Each prediction model, its capabilities and description of outputs is summarized in Table 9.5-2.

9.5.1.1 In-house Historical Data Collection Model

A few organizations predict software reliability by collecting and using the database of information accumulated on each of their own software projects. Metrics employed include Product, Project Management, and Fault indicators. Statistical regression analysis typically is used to develop a prediction equation for each of the important project characteristics. Management uses this information to predict the reliability of the proposed software product as well as to plan resource allocation.

TABLE 9.5-2: SOFTWARE RELIABILITY PREDICTION TECHNIQUES

Prediction Model	Capabilities	Description of Outputs
Historical Data Collection Model	Can be most accurate, if there is organization wide commitment.	Produces a prediction of the failure rate of delivered software based on company wide historical data.
Musa's Model	Predicts failure rate at start of system test that can be used later in reliability growth models.	Produces a prediction of the failure rate at the start of system test.
Putnam's Model	The profile of predicted faults over time and not just the total number is needed. Can be used with the other prediction models.	Produces a prediction in the form of a predicted fault profile over the life of the project.
TR-92-52 Model	Allows for tradeoffs.	Produces a prediction in terms of fault density or estimated number of inherent faults.
TR-92-15 Model	Has default factors for estimating number of faults	Estimates faults during each development phase.

9.5.1.2 Musa's Execution Time Model

Developed by John Musa (Ref. [12]) of Bell Laboratories in the mid 1970s, this was one of the earliest reliability prediction models. It predicts the initial failure rate (intensity) of a software system at the point when software system testing begins (i.e., when time, t = 0). The *initial failure intensity*, λ_0, (faults per unit time) is a function of the unknown, but estimated, total number of failures expected in infinite time, N. The prediction equation is shown below; terms are explained in Table 9.5-3.

$$\lambda_0 = k \times p \times w_o$$

For example, a 100 line (SLOC) FORTRAN program with an average execution rate of 150 lines per second has a predicted failure rate, when system test begins, of $\lambda_0 = k \times p \times w_0$ = (4.2E-7) x (150/100/3) x (6/1000) = .0126E-7 = 1.26E-9 faults per second (or 1 fault per 7.9365E8 seconds which is equivalent to 1 fault per 25.17 years).

It is important to note that this time measure is *execution time*, not calendar time. Since hardware reliability models typically are in terms of calendar time, it is not feasible to use Musa's prediction in developing an overall system reliability estimate unless one is willing to assume that calendar time and execution time are the same (usually not a valid assumption).

TABLE 9.5-3: TERMS IN MUSA'S EXECUTION TIME MODEL

Symbol	Represents	Value
k	Constant that accounts for the dynamic structure of the program and the varying machines	k = 4.2E-7
p	Estimate of the number of executions per time unit	p = r/SLOC/ER
r	Average instruction execution rate, determined from the manufacturer or benchmarking	Constant
SLOC	Source lines of code (not including reused code)	
ER	Expansion ratio, a constant dependent upon programming language	Assembler, 1.0; Macro Assembler, 1.5; C, 2.5; COBAL, FORTRAN, 3; Ada, 4.5
w_0	Estimate of the initial number of faults in the program	Can be calculated using: w_0 = N x B or a default of 6 faults/1000 SLOC can be assumed
N	Total number of inherent faults	Estimated based upon judgment or past experience
B	Fault to failure conversion rate; proportion of faults that become failures. Proportion of faults not corrected before the product is delivered.	Assume B = .95; i.e., 95% of the faults undetected at delivery become failures after delivery

9.5.1.3 Putnam's Model

Trachtenberg (formerly of General Electric) and Gaffney (of then IBM Federal Systems, now Loral) examined defect histories, by phases of the development process, for many projects of varying size and application type. Based on their work, Putnam (Ref. [5]) assigned the general normalized Rayleigh distribution to describe the observed reliability, where k and a are constants fit from the data and t is time, in months:

$$R(t) = k \exp(-at^2)$$

The corresponding probability density function, f(t), the derivative of R(t) with respect to t, is of the general form:

$$f(t) = 2ak\, t \exp(-at^2)$$

Putnam further developed an ordinal (i.e., not equally spaced in real time) scale to represent the

development process milestones; see Table 9.5-4. Of special interest is Milestone 7, denoted by t_d, corresponding to the end of the development phases and the beginning of full operational capability; this point was defined as occurring at the 95th percentile (i.e., 95% of all defects have been detected at this point in the software development). Using t_d as the reference basis, he then developed the expressions for the model constants, a and k, in terms N and t_d. The final equation to predict the expected number of defects per month as a function of the schedule month and the total number of inherent defects, N, is given by:

$$f(t) = (6N/t_d^2)\, t \exp(-3t^2/t_d^2)$$

TABLE 9.5-4: PUTNAM'S TIME AXIS MILESTONES

Milestone #	Milestone
0	Feasibility study
1	Preliminary design review, function design complete
2	Critical design review, detailed design complete
3	First code complete
4	Start of system integration test
5	Start of user systems test
6	Initial operational capability; installation
7	Full operational capability; reliability about 95% in routine usage
8	99% reliability achieved by stress testing
9	99.9% reliability, assumed debugged

For example, suppose a FORTRAN program is being developed; the plan is that it will be fully operational (Milestone 7) in 10 calendar months resulting in t_d^2 to be 10^2 or 100. The defects expected per month during development are calculated using the expression:

$$f(t) = .06\, N\, t \exp(-.03t^2)$$

Calculation results are shown in Figure 9.5-1, where t is the month number, f(t) is the expected proportion of the total number of defects to be observed in month t, and F(t) represents the cumulative proportion. The Milestone number, based on the planned development schedule is also shown for comparison; Milestone 7, corresponding to the 95th percentile, is, indeed, in Month 10, Milestone 8, at the 99th percentile, is expected to occur in scheduled Month 13, and Milestone 9, at .999, is not expected to be reached by the end of scheduled Month 15.

t	f(t)	F(t)	Mile #
1	0.058	0.058	
2	0.106	0.165	1
3	0.137	0.302	
4	0.149	0.451	2
5	0.142	0.592	
6	0.122	0.715	3
7	0.097	0.811	4
8	0.070	0.881	5
9	0.048	0.929	6
10	0.030	0.959	7
11	0.017	0.976	
12	0.010	0.986	
13	0.005	0.991	8
14	0.002	0.993	
15	0.001	0.994	

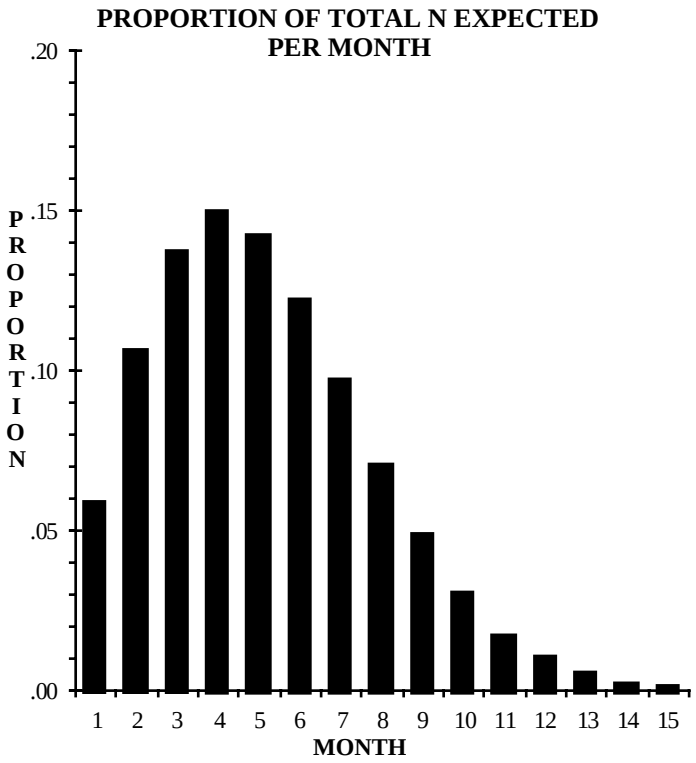


FIGURE 9.5-1: EXPECTED PROPORTION OF THE TOTAL NUMBER OF DEFECTS

One major benefit of this model is that the expected number of faults can be predicted for various points in development process as compared to Musa's model that provides the prediction when system testing begins (i.e., at Milestone 4) only.

Another corollary to this model is that the mean time to the next defect (MTTD) is given by 1/f(t). This is only meaningful after Milestone 4 (since prior to that point the system would not have been developed so defects could not be detected). As the development progresses, (i.e., t increases), the MTTD increases since defects are being eliminated.

9.5.1.4 Rome Laboratory Prediction Model: RL-TR-92-52 (Ref. [16])

This is a method for predicting *fault density at delivery time* (i.e., at Putnam's Milestone 6) and subsequently using this fault density to predict the *total number of inherent faults*, N, and the *failure rate*. It also provides a mechanism for allocating software reliability as a function of the software characteristics as well as assessing trade-off options. The basic terminology of this model is presented in Table 9.5-5. The underlying assumption is that Source Lines of Code (SLOC) is a valid size metric.

TABLE 9.5-5: RL-TR-92-52 TERMINOLOGY

Terms	Description
A	Factor selected based on Application type; represents the baseline fault density
D	Factor selected to reflect the Development environment
S	Factor calculated from various "sub-factors" to reflect the Software characteristics
SLOC	The number of executable Source Lines Of Code; lines that are blank or contain comments to enhance the readability of the code are excluded
FD	Fault density; for the purposes of this model it is defined as the ratio of faults to lines of code (faults/SLOC)
N	Estimate of total number of inherent faults in the system; prediction is derived from the fault density and the system size
C	Factor representing a Conversion ratio associated with each application type; values are determined by dividing the average operational failure rate by the average fault density in the baseline sample set.

It is recognized as one of a few publicly available prediction models based upon extensive historical information. Predictions are based on data collected on various types of software systems developed for the Air Force; see Table 9.5-6.

TABLE 9.5-6: AMOUNT OF HISTORICAL DATA INCLUDED

Application Type	# of Systems	Total SLOC
Airborne	7	540,617
Strategic	21	1,793,831
Tactical	5	88,252
Process Control	2	140,090
Production Center	12	2,575,427
Developmental	6	193,435
TOTAL	53	5,331,652

The basic equations are:

Fault Density = FD = A x D x S (faults/line)
Estimated Number of Inherent Faults = N = FD x SLOC
Failure Rate = FD x C (faults/time)

The model consists of factors that are used to predict the fault density of the software application. These factors are illustrated in Table 9.5-7.

TABLE 9.5-7: SUMMARY OF THE RL-TR-92-52 MODEL

Factor	Measure	Range of Values	Phase Used In*	Trade-off Range
A - Application	Difficulty in developing various application types	2 to 14 (defects/KSLOC)	A-T	None - Fixed
D - Development organization	Development organization, methods, tools, techniques, documentation	.5 to 2.0	If known at A, D-T	The largest range
SA - Software anomaly management	Indication of fault tolerant design	.9 to 1.1	Normally, C-T	Small
ST - Software traceability	Traceability of design and code to requirements	.9 to 1.0	Normally, C-T	Large
SQ - Software quality	Adherence to coding standards	1.0 to 1.1	Normally, C-T	Small
SL - Software language	Normalizes fault density by language type	Not applicable	C-T	N/A
SX - Software complexity	Unit complexity	.8 to 1.5	C-T	Large
SM - Software modularity	Unit size	.9 to 2.0	C-T	Large
SR - Software standards review	Compliance with design rules	.75 to 1.5	C-T	Large

Key **A - Concept or Analysis Phase**

D - Detailed and Top Level Design
C - Coding
T - Testing

The following are benefits of using this model:

(1) It can be used as soon as the concept of the software is known

(2) During the concept phase, it allows “what-if” analysis to be performed to determine the impact of the development environment on fault density

(3) During the design phase, it allows “what-if” analysis to be performed to determine the impact of software characteristics on fault density

(4) It allows for system software reliability allocation because it can be applied uniquely to each application type comprising a software system

(5) The prediction can be customized using unique values for the A, S, and D factors based upon historical software data from the specific organization's environment while the following are drawbacks:

(a) Factors and values used were generated based on software developed for the Air Force; if the software in question does not match one of the Air Force-related application types, then the average value must be selected. The Air Force application types do not map well to software developed outside the military environment

(b) Use of SLOC as the size metric is becoming more and more irrelevant with recent changes in software development technology, such as Graphical User Interface (GUI) system development, and the use of Commercial Off-the-Shelf (COTS) software

9.5.1.5 Rome Laboratory Prediction Model: RL-TR-92-15 (Ref. [17])

This technical report, produced by Hughes Aircraft for Rome Laboratory, examined many software systems. It resulted in an *average fault rate prediction* value of 6 faults/1000 SLOC. (This was the default value for fault rate, w_0, used in Musa's Execution Time Model).
In addition, a set of 24 predictor factors, listed in Table 9.5-8, was used to estimate the three main variables of interest:

(1) Number of faults detected during each development phase (DP)

(2) Man-hours utilized during each phase (UT)

(3) Size of product (S)

The resultant equations were:

(1) $f(DP) = 18.04 + .05 \times (.009\,X_1 + .99\,X_2 + .10\,X_3 - .0001\,X_4 + .0005\,X_5)$

(2) $f(UT) = 17.90 + .04 \times (.007\,X_1 + .796\,X_2 + .08\,X_3 - .0003\,X_4 + .0003\,X_5 + .00009\,X_6 + .0043\,X_7 + .013\,X_8 + .6\,X_9 + .003\,X_{10})$

(3) $f(S) = 17.88 + .04 \times (.0007\,X_1 + .8\,X_3 + .01\,X_8 + .6\,X_9 + .008\,X_{23} + .03\,X_{25})$

where the coefficients and descriptions of the variables of the regression model are listed in the table.

TABLE 9.5-8: REGRESSION EQUATION COEFFICIENTS

X	Description of Variable	Coefficients		
		EQ 1	EQ 2	EQ 3
1	Number of faults in software requirements specification	.009	.007	.007
2	Requirements statement in specification	.99	.796	NA
3	Pages in specification	.10	.08	.80
4	Man-months spent in requirements analysis	.0001	-.0003	NA
5	Requirements change after baseline	.0005	.0003	NA
6	Number of faults in preliminary design document	NA	.00009	NA
7	Number of CSCS	NA	.0043	NA
8	Number of units in design	NA	.013	.01
9	Pages in design document	NA	.6	.6
10	Man-months spent in preliminary design	NA	.003	NA
11	Number of failures in design document	NA	NA	NA
12	Man-months spent in detailed design	NA	NA	NA
13	Design faults identified after baseline	NA	NA	NA
14	Design faults identified after internal review	NA	NA	NA
15	Number of executable SLOC	NA	NA	NA
16	Faults found in code reviews	NA	NA	NA
17	Average years of programmer experience	NA	NA	NA
18	Number of units under review	NA	NA	NA
19	Average number of SLOC per unit	NA	NA	NA
20	Average number branches in unit	NA	NA	NA
21	Percentage branches covered	NA	NA	NA
22	Nesting depth coverage	NA	NA	NA
23	Number of times an unit is unit tested	NA	NA	.008
24	Man-months for coding and unit test	NA	NA	NA
25	Equals (X13 + X14 + X16)	NA	NA	.03

The results indicate that thirteen of the 24 hypothesized factors had no effect on the three variables of interest. Further, the most important estimators involved the software requirements specification, including the number of requirement statements, number of faults in these statements, and the total number of pages in the specification.

The benefits of this model are:

(1) It can be used prior to system testing to estimate reliability

(2) It includes cost and product parameters as well as fault and time

The disadvantages of this model are:

(1) It was based on data collected by one organization in one industry/application type

(2) It does not disclose the unit of measure for specification size

9.5.2 Estimation Models

The fault count and fault rate models are the most common type of estimation techniques. Each makes assumption about how faults arrive (detected) and how they are corrected. The fault count models include: Exponential, Weibull and Bayesian techniques. Also, included in the estimation model scenario are the test coverage and fault tagging methods.

9.5.2.1 Exponential Distribution Models

In general, *exponential models* assume that the software is in an operational state and that all faults are independent of each other. The time to failure, t, of an individual fault follows the exponential distribution:

$$f(t) = \lambda \exp(-\lambda t)$$

with the general form for the reliability given by:

$$R(t) = \exp(-\lambda t)$$

and the mean time to the next failure (MTTF) expressed as:

$$MTTF = 1/\lambda$$

The notations used in the general case for the exponential distribution model are shown in Table 9.5-9 and illustrated in Figure 9.5-2.

TABLE 9.5-9: NOTATIONS FOR THE EXPONENTIAL DISTRIBUTION MODEL

Notation	Explanation
N	Total number of defects
n	Number of defects to date
c	Number of defects corrected to date
N-n	Defects not yet manifested
N-c	Defects yet to be corrected
n_f	Fault count
λ_f	Fault rate
t_f	Future time
n_p	Fault count at present time
λ_p	Fault rate at present time
t_p	Present time

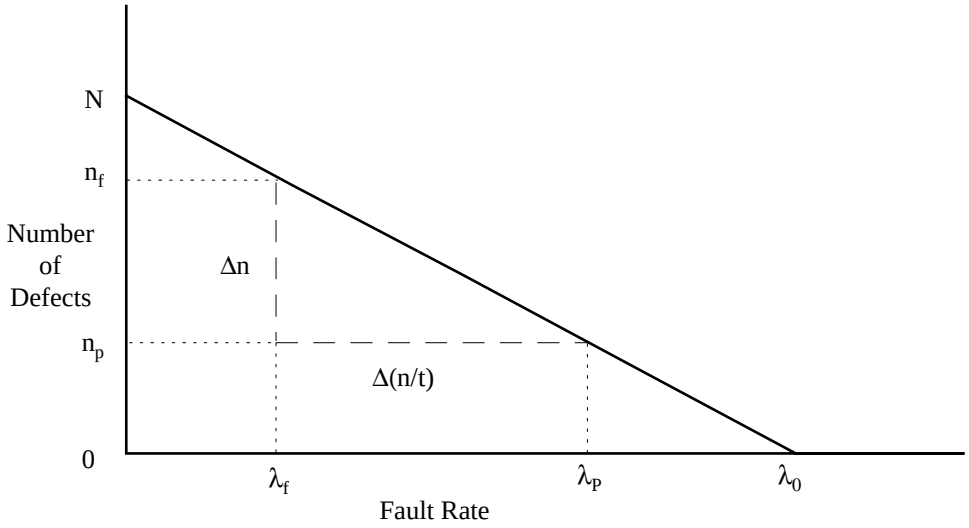


FIGURE 9.5-2: EXPONENTIAL MODEL BASIS

Advocates of the exponential model note its simplicity and its parallelism to the hardware reliability framework. The major disadvantage is that it cannot be used early in the software development since the product must be operational before the model can be used. Hence, it cannot be used for early reliability assessment.

Table 9.5-10 summarizes the various exponential models including assumptions and comments.

TABLE 9.5-10: VARIOUS EXPONENTIAL MODELS*

Model	MTTF	Dn	Dt	Assumptions/Comments
General Exponential	$1/[k(N - c)]$	$k^{-1} \lambda_p / \lambda_f$	$k^{-1} ln (\lambda_p / \lambda_f)$	Faults are equal in severity and probability of detection Fault rate directly related to number of faults remaining to be corrected
Lloyd-Lipow	$1/[k(N - n)]$	$k^{-1} \lambda_p / \lambda_f$	$k^{-1} ln (\lambda_p / \lambda_f)$	Fault rate directly related to number of faults remaining to be detected
Musa's Basic		$N/\lambda_0 (\lambda_p - \lambda_f)$	$N/\lambda_0 ln (\lambda_p - \lambda_f)$	References an initial fault rate at time 0 (beginning of system test)
Musa's Logarithmic		$f^{-1} ln (\lambda_p / \lambda_f)$	$f^{-1} (1/\lambda_f - 1/\lambda_p)$	Some faults are more likely to be found before others Rate of fault detection decreases exponentially
Shooman's	$1/[kSLOC - ((N/SLOC) - (C/SLOC))]$	$k^{-1} \lambda_p / \lambda_f$	$k^{-1} ln (\lambda_p / \lambda_f)$	Adjusts for changing product size; each parameter is normalized for lines of code
Goel-Okumoto				Faults can cause other faults Faults may not be removed immediately

* where k is a constant of proportionality, N is the number of inherent faults, c is the number of corrected faults and n is the number of detected faults

General Exponential Model

In the general case, the model assumes that all faults are equal in severity and probability of detection and that each is immediately corrected upon detection. The fault rate, λ, is assumed to be directly related to the number of faults remaining in the software. That is, λ is a function of the number of corrected faults, c:

$$\lambda = k (N - c)$$

where k is a constant of proportionality. In actual application, k is typically estimated from the slope of the plot of the observed fault rate vs. number of faults corrected.

The projection of the number of faults needed to be detected to reach a final failure rate λ_f is given by:

$$\Delta n = (1/k) \lambda_p / \lambda_f$$

where k is the same proportionality constant used above.

The projection of the time necessary to reach a projected fault rate is given by:

$$\Delta t = (1/k)\ln\left[\lambda_p/\lambda_f\right]$$

The major disadvantage of this specific approach is that not only must the defects be detected but they also must be corrected.

Lloyd-Lipow Model (Ref. [18] and [19])

The *Lloyd-Lipow Model* exponential model also assumes that all faults are equal in severity and probability of detection. The difference from the previous model is that in this Lloyd-Lipow approach, the fault rate, λ, is assumed to be directly related to the number of faults remaining to be **detected** (not corrected) in the software. That is, λ is a function of the number of detected faults, n:

$$\lambda = k\,(N - n)$$

The expressions for the mean-time-to-failure (MTTF), Δn and Δt are the same as in the general exponential model.

This form of the exponential model does not require defect correction, just detection. However, the validity of the use of the exponential model in this situation has been questioned.

Musa's Basic Model (Ref. [12])

Musa's Basic Model is another form of the general exponential model. It utilizes the initial (i.e., at the start of software testing) fault rate, λ_0, where either λ_0 is estimated from the data or computed ($\lambda_0 = N/k$) based on a guess for N and the estimate for k, the previously referenced slope value.

In this model, the fault rate after n faults have been detected is a fraction of the original fault rate:

$$\lambda_n = \lambda_0\,(1 - n/v)$$

where:

n is usually expressed as μ and v is usually expressed as υ

while the expression for the fault rate at time t is given by:

$$\lambda_t = \lambda_0 \exp\left[-(\lambda_0/\upsilon)\tau\right]$$

where:

υ = N/B, where N is the number of inherent faults and B is the fault reduction ratio, usually assumed to be 95% (i.e., 95% of the faults undetected at delivery become failures after delivery

τ = System test time

The projection of the number of faults needed to be detected to reach a final failure rate λ_f is given by:

$$\Delta n = N/\lambda_0 \, (\lambda_p - \lambda_f)$$

The projection of the time necessary to reach a projected failure rate is given by:

$$\Delta t = N/\lambda_0 \, \ln (\lambda_p - \lambda_f)$$

The disadvantage is that this model is very sensitive to deviations from the assumptions. In addition, as noted with Musa's previous work, the units are execution time, not calendar time.

Musa's Logarithmic Model (Ref. [12])

Musa's Logarithmic Model has different assumptions than the other exponential models:

(1) Some faults are likely to be found before others

(2) The rate of fault detection is not constant, but decreases exponentially

In this model, the fault rate after n faults have been detected is a function of the original fault rate:

$$\lambda_n = \lambda_0 \exp(-ft)$$

while the expression for the fault rate at time t is given by:

$$\lambda_t = \lambda_0/(\lambda_0 \, f \, t + 1)$$

where:

f = failure intensity decay parameter, the relative change of n/t over n.

The projection of the number of faults needed to be detected to reach a final failure rate λ_f is given by:

$$\Delta n = 1/f \ln (\lambda_p/\lambda_f)$$

The projection of the time necessary to reach a projected failure rate is given by:

$$\Delta t = 1/f (1/\lambda_f - 1/\lambda_p)$$

The major benefit of this model is that it does not require an estimate for N. Since the value for f can be estimated prior to actual data occurrence, the model can be used earlier in the development cycle to estimate reliability.

The disadvantage of this model, typical for most exponential models, is that the model assumptions must be valid for the results to be valid. In particular, the assumption that the rate of fault detection decreases exponentially has not been confirmed with many real data sets. In addition, as noted with Musa's previous work, the units are execution time, not calendar time, making direct comparison with hardware reliability difficult.

Shooman's Model (Ref. [20])

The *Shooman's Model* is similar to the general exponential model except that each fault count is normalized for the lines of code at that point in time. Earlier, $\lambda = k (N - c)$; here, it is given by:

$$\lambda = k \, SLOC (N/SLOC - c/SLOC)$$

The equation of $MTTF = 1/\lambda$ uses Shooman's expression for λ. The equations for Δn and Δt are the same as the general exponential case.

The advantage of Shooman's model is that it adjusts for the changing size of the software product. The disadvantages are that it must be used later in development after the LOC have been determined and that the general exponential assumptions may not apply.

Goel-Okumoto Model (Ref. [19])

This model is different from other exponential models because it assumes that faults can cause other faults and that they may not be removed immediately. An iterative solution is required. This model is expressed as:

$$\lambda_t = ab \exp(-bt)$$

where a and b are resolved iteratively from the following:

$$n/a = 1 - \exp(-bt) \quad \text{and} \quad n/b = a\, t \exp(-bt) + \sum t_i ,$$

where the summation is over i = 1,..., n,

Use N and k as starting points for solving for these two equations simultaneously.

The major benefit of this model is that it can be used earlier than other exponential models while its major disadvantage is that it is very sensitive to deviations from the assumptions.

9.5.2.2 Weibull Distribution Model (Ref. [19])

The *Weibull Model* is one of the earliest models applied to software. It has the same form as that used for hardware reliability. There are two parameters: a, the scale parameter (a > 0), and b, the shape parameter that reflects the increasing (b > 1), decreasing (b < 1) or constant (b = 1) failure rate.

The mean time to next failure is given by:

$$MTTF = (b/a)\,\Gamma(1/a)$$

where $\Gamma(c)$ is the complete Gamma Function = $\int_0^\infty y^{c-1} e^{-y}\, dy$

The reliability at time t is given by:

$$R(t) = \exp[-(t/b)^a]$$

The benefits of the Weibull model is its flexibility to take into account increasing and decreasing failure rates. The disadvantage of this model is more work is required in estimating the parameters over the exponential model.

9.5.2.3 Bayesian Fault Rate Estimation Model

The *Bayesian* approach does not focus on the estimated inherent fault count, N, but rather concentrates on the fault/failure rate. The classical approach assumes that reliability and failure rate are a function of fault detection while the Bayesian approach, on the other hand, assumes that a software program which has had fault-free operation is more likely to be reliable. The Bayesian approach also differs because it is possible to include an assessment of “prior knowledge” (therefore, it is sometimes called a “subjective” approach).

The Thompson and Chelson's model (Ref. [19]) assumes that:

(1) Software is operational

(2) Software faults occur at some unknown rate λ that is assumed to follow a Gamma Distribution with parameters X_i and $f_i + 1$

(3) Faults are corrected in between test periods but not during test periods

(4) Total number of faults observed in a single testing period of length t_i follows a Poisson distribution with parameter λt_i

The model assumes that there are i test periods, each with length, t_i (not assumed equal); where the number of faults detected during that period is represented by f_i. The subjective information is inserted as occurring in period 0, i.e., t_0 and f_0 represent the prior information. If there is no prior information, these values are set to zero. On the other hand, if there is a great deal of experience, t_0 might be very large, especially relative to the expected evaluation time; the value for the prior number of faults also depends on past experience and is independent from the prior of time.

Let T_i represent the cumulative total of the test period lengths over the entire range, i.e., from period 0 to i and let F_i represent the cumulative total of the faults, f_i, over the entire range, i.e., from period 0 to i.

Then, the reliability at time t (in interval i) is expressed as a function of the values from the previous (i - 1) interval as well as the current i^{th} interval data:

$$R(t) = [T_{i-1}/(T_{i-1} + t)]^{F_{i-1}}$$

The failure rate estimate at time t (in interval i) is given by:

$$\lambda(t) = (F_{i-1} + 1)/T_{i-1}$$

The benefits of this model are related to its assumptions: N is not assumed fixed, reliability is not assumed to be directly a function of N, and faults are not assumed to be corrected immediately. The disadvantage is that Bayesian Models, in general, are not universally accepted since they allow for the inclusion of prior information reflecting the analyst's degree of belief about the failure rate.

9.5.2.4 Test Coverage Reliability Metrics

Test coverage advocates have defined software reliability as a function of the amount of the software product that has been successfully verified or tested. Three such metrics are discussed below. The first is a simple ratio based upon the rate of successful testing during the final acceptance test. The second and third provide a metric based on ways of combining the results from both white-box and black-box testing.

Advocates of this approach to reliability explain that since the data are (or should be) collected and tracked during testing, these measures are readily available and require no additional verification effort. However, to the reliability engineer, these metrics are foreign to anything used in the hardware environment to describe reliability. Further, none of these metrics can be converted to failure rate or used to predict or estimate mean time between failures.

Test Success Reliability Metric

In this approach, (Ref. [19]) reliability is simply defined as the ratio of the number of test cases executed successfully during acceptance (black-box) testing, defined as s, to the total number of test cases executed during acceptance testing, defined as r:

$$R = s/r$$

The validity of the result is dependent on the size of r as well as the ability for r to represent the total operational profile of the software. During the very late stages of testing, immediately prior to delivery, this model may be used for accepting or rejecting the software.

IEEE Test Coverage Reliability Metric

This method (Refs. [21] and [22]) assumes that *reliability is dependent upon both the functions that are tested (black-box) and the product that is tested (white-box).* It assumes that both types of testing have to be completed for the test coverage to be complete. The reliability value is defined as the product of two proportions, converted to a percent:

$$R = p(\text{functions tested}) * p(\text{program tested}) * 100\%$$

where:

p(functions tested) = Number of capabilities tested/total number of capabilities

p(program tested) = Total paths and inputs tested/total number of paths and inputs

Leone's Test Coverage Reliability Metric

This approach (Ref. [23]) is similar to the IEEE Model except that it *assumes that it is possible to have either white or black box testing and still have some level of reliability.* Two white-box

variables, a and b, and two black-box variables, c and d, are assessed. The reliability is the weighted sum of the four proportions:

$$R = ((a * w1) + (b * w2) + (c * w3) + (d * w4)) / (w1 + w2 + w3 + w4)$$

where:

a = Number of independent paths tested/total number of paths
b = Number of inputs tested/total number of inputs
c = Number of functions verified/total number of functions
d = Number of failure modes addressed/total number of failure modes

The values for w1, w2, w3, w4 represent weights. If all parameters are equally important, these weights all are set to 1; however, if there are data to support that some parameters are more important than others, then these more important parameters would receive higher weights.
This model has two underlying assumptions. First, independent paths are identified using information from testing procedures. Second, failure models (Ref. [24]) are identified using Fault Tree Analysis or Failure Modes, Effects, and Criticality Analysis.

9.5.3 Estimating Total Number of Faults Using Tagging

Tagging (Ref. [23]) is used to estimate the total number of faults in the software, N, based on the number observed during testing. It is based on *seeding*, a method of introducing faults into the software and then determining how many of these faults are found during testing in order to estimate the total number of faults.

To illustrate, suppose the number of fish in a pond, N, is of interest. One way to develop an estimate is to capture and tag some number of the fish, T, and return them to the pond. As fish are caught, the number of tagged fish, t, is recorded as well as the number untagged, u. The total number of untagged fish U, is estimated using the proportion: u/U = t/T. Then, the total number of fish is estimated as the sum: N = U + T.

The steps used in the basic seeding approach for a software fault estimation are:

(1) A set of faults which represents the faults which would typically be found in operational usage is identified.

(2) Faults are injected into software without the testers or developers being aware of them. The total number of injected faults is T.

(3) Test software and identify all faults found. Let t = the number of faults detected that were injected and let u = the number of faults detected which were not injected.

(4) The total number of faults which are not injected is estimated by U, where:

$$u/U = t/T$$

(5) The total number of faults, N, is estimated by:

$$N = U + T$$

(6) The injected faults are removed.

In general, this approach is not recommended based upon the issues identified below:

(1) How can faults which are typical of operational usage be identified and then injected in a completely random manner? (without bias)?
(2) Seeding assumes faults are due to coding mistakes. What about faults due to requirements, design and maintenance errors?
(3) During the course of a typical testing cycle, faults are typically corrected. Do the injected faults get corrected during this process, or at the very end.
(4) Will seeded faults prevent real faults from being detected?
(5) How can injected faults be kept a secret when the maintainer goes to fix them? How can it be justified to spend resources fixing injected faults?
(6) What is the action at the end of testing, go back to the original version with no injected faults (and no corrected real faults) or remove (hopefully all) the injected faults?

An alternative *Dual Test Group Approach* is similar to basic seeding except that two groups are used. It assumes that:

(1) Two independent test groups are testing the same software at same time.
(2) Groups do not share information on faults detected in testing.
(3) Groups create their own test plans, but test the same functionality of the software.
(4) Groups are equal in experience and capabilities.

This model predicts N, the total number of faults, based upon three numbers, n_1, n_2, and n_{12} where n_1 and n_2 represent the number of faults found by Group 1 and Group 2, respectively, while n_{12} is the number of faults found by both groups.

The total number of faults, N, is estimated by:

$$N = R + n_1 + n_2 + n_{12}$$

where:
R is the estimated total number of remaining faults

This model assumes that as the number of faults found by both groups increases, the number remaining decreases. As testing continues, it is assumed that n_{12} will increase. This means that when there are few faults left in the software (i.e., as R approaches 0), both test groups will begin finding the same faults. This may not be the case, however, since both test groups may be inefficient. The basic assumptions also may be unrealistic. It is not always possible or economical to have two completely independent test groups with equal experience level and capabilities. It also may not be easy to keep the groups independent and equal in experience.

9.6 Software Reliability Allocation

Software reliability allocation involves the establishment of reliability goals for individual computer software configuration items (CSCI) based on top-level reliability requirements for all the software. It is very important that this activity, allocations, be established early in the program so that criteria for evaluating the achieved reliability of each element can be established. Table 9.6-1 describes five allocation techniques. These techniques are based on the type of execution expected or the operational profile or the software complexity.

The allocation of a system requirement to software elements makes sense only at the software system or CSCI level. Once software CSCIs have been allocated reliability requirements, a different approach is needed to allocate the software CSCI requirements to lower levels. The reliability model for software differs significantly from hardware due to its inherent operating characteristics. For each mode in a software system's (CSCI) operation, different software modules (CSCs) will be executing. Each mode will have a unique time of operation associated with it. A model should be developed for the software portion of a system to illustrate the modules which will be operating during each system mode, and indicate the duration of each system mode. An example of this type of model is shown in Table 9.6-2 for a missile system.

TABLE 9.6-1: SOFTWARE RELIABILITY ALLOCATION TECHNIQUES (REF. [2])

Technique	Procedure Name	Use Description
Sequential Execution (see 9.6.1)	Equal apportionment applied to sequential software CSCIs	Use early in the SW development process when the software components are executed sequentially
Concurrent Execution (see 9.6.2)	Equal apportionment applied to concurrent software CSCIs	Use early in the SW development process and the software components are executed concurrently
Operational Profile (see Ref. [2])	Mission or Operational Profile Allocation	Use when the operational profile of the CSCIs are known
Operational Criticality (see 9.6.3)	Allocation based on operational criticality factors	Use when the operational criticality characteristics of the software is known
Complexity (see 9.6.4)	Allocation based on complexity factors	Use when the complexity factors of the software components are known

The software reliability model will include the number of source lines of code (SLOC) expected for each module. These data, along with other information pertaining to software development resources (personnel, computing facilities, test facilities, etc.) are used to establish initial failure intensity predictions for the software modules.

To assist in the proper selection of an allocation technique, a flow diagram is provided in Figure 9.6-1.

TABLE 9.6-2: SOFTWARE FUNCTIONS BY SYSTEM MODE - EXAMPLE

System Mode	Modules	SLOC
Standby - 2 Hours	Built-in Test (BIT)	4000
	1760 Interface	750
	Flight Sequencing	2000
	Prelaunch Initialization	900
	TOTAL	**7650**
Prelaunch - 20 Minutes	BIT	4000
	Navigation	1000
	Flight Sequencing	2000
	Prelaunch Initialization	900
	Navigation Kalman Filter	2000
	TOTAL	**9900**
Post-Launch - 10 Minutes	BIT	4000
	Interface	7000
	Navigation	1000
	Infrared Seeker Control	500
	Flight Sequencing	2000
	Terminal Maneuver	1000
	Other Post-Launch	24500
	Navigation Kalman Filter	2000
	TOTAL	**42000**

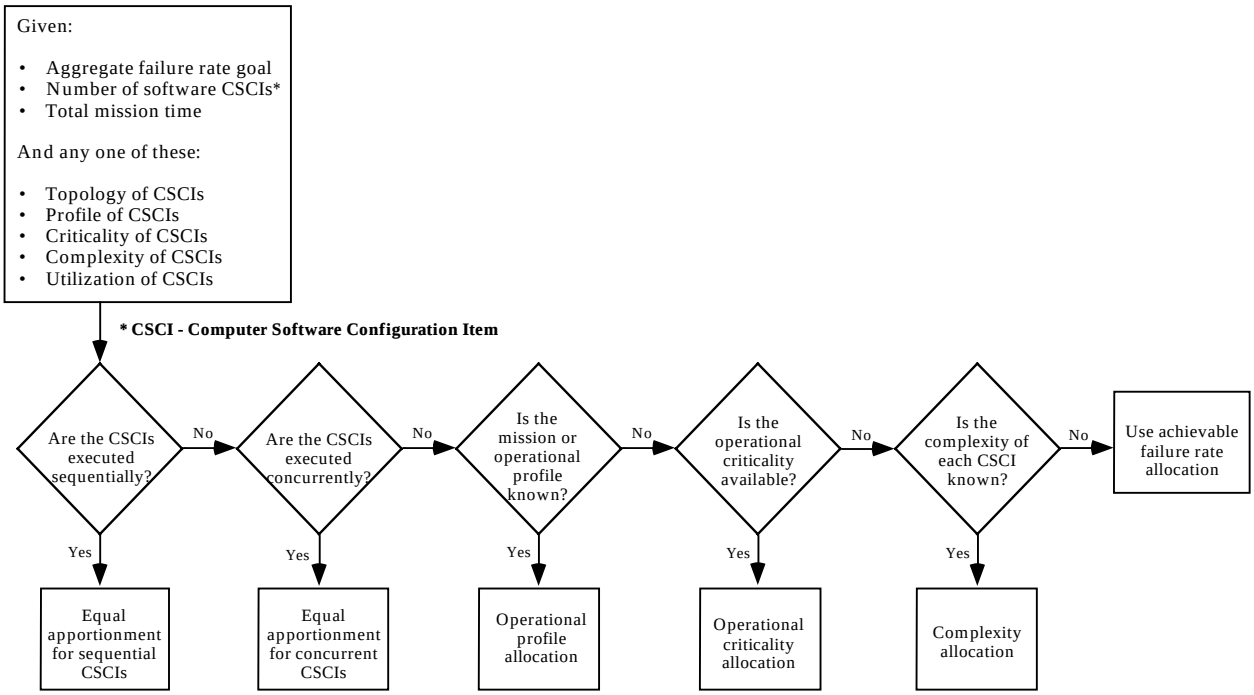


FIGURE 9.6-1: RELIABILITY ALLOCATION PROCESS (REF. [2])

9.6.1 Equal Apportionment Applied to Sequential Software CSCIs

This technique is used to allocate a failure rate goal to each individual software CSCI when the CSCIs are executed sequentially. This procedure should be used only when the failure rate goal of the software aggregate (λ_S), and the number of software CSCIs in the aggregate (N), are known. The aggregate's failure rate goal is either specified in the requirements or is the result of an allocation performed at a higher level in the system hierarchy.

Steps:

(1) Determine the failure rate goal for the software aggregate; λ_S

(2) Determine the number of software CSCIs in the aggregate; N

(3) For each software CSCI, assign the failure rate goal as follows:

$$\lambda_{i\,(CSCI)} = \lambda_s \text{ (failures per hour)}$$

where:
i = 1, 2, ..., N

Example:

A software aggregate is required to have a maximum of 0.05 failures per hour. The aggregate consists of five software CSCIs that are executed one after another, that is, the five CSCIs run sequentially. All CSCIs must succeed for the system to succeed (this is a series system).

Then, using the equal apportionment technique, the failure rate goal for the i^{th} software CSCI is assigned to be:

$\lambda_i = \lambda_S = 0.05$ failures per hours

where:
i = 1, 2, ..., 5

9.6.2 Equal Apportionment Applied to Concurrent Software CSCIs

This technique is used to allocate the appropriate failure rate goal to each individual software CSCI, when the CSCIs are executed concurrently. λ_S, the failure rate of the software aggregate, and N, the number of software CSCIs in the aggregate, are needed for this procedure.

Steps:

(1) Determine the failure rate goal for the software aggregate; λ_S

(2) Determine the number of software CSCIs in the aggregate; N

(3) For each software CSCI, assign the failure rate goal as follows:

$$\lambda_{i\,(CSCI)} = \lambda_s / N \text{ (failures per hour)}$$

where:
i = 1, 2, ..., N

Example:

A software aggregate has a failure rate goal of 0.05 failures per hour. The aggregate consists of five software CSCIs, which are in series and executed concurrently. Then, the allocated failure rate goal of each of the five software CSCI is:

$$\lambda_i = \lambda_s / N = \frac{0.05}{5} = 0.01 \text{ failures per hour}$$

9.6.3 Allocation Based on Operational Criticality Factors

The operational criticality factors method allocates failure rates based on the system impact of a software failure. Criticality is a measure of the system's ability to continue to operate and the system's ability to be fail-safe. For certain modes of operation, the criticality of that mode may call for a lower failure rate to be allocated. In order to meet very low failure rates, fault-tolerance or other methods may be needed.

The following procedure is used to allocate the appropriate value to the failure rate of each software CSCI in an aggregate, provided that the criticality factor of each CSCI is known. A CSCI's criticality refers to the degree to which the reliability and/or safety of the system as a whole is dependent on the proper functioning of the CSCI. Furthermore, gradations of safety hazards translate into gradations of criticality. The greater the criticality, the lower the failure rate that should be allocated.

Steps:

(1) Determine the failure rate goal of the software aggregate; λ_S

(2) Determine the number of software CSCIs in the aggregate; N

(3) For each i^{th} CSCI, i = 1, 2, ..., N, determine its criticality factor c_i. The lower the c_i the more critical the CSCI.

(4) Determine τ_i' the total active time of the i^{th} CSCI, i = 1, 2, ..., N. Determine T, the mission time of the aggregate.

(5) Compute the failure rate adjustment factor K:

$$K = \frac{\sum_{i=1}^{N} c_i \tau_i'}{T}$$

(6) Compute the allocated failure rate goal of each CSCI

$$\lambda_i = \lambda_s (c_i / K)$$

(Dividing by K makes the allocated CSCI failure rates build up to the aggregate failure rate goal).

Example:

Suppose a software aggregate consisting of three software CSCIs is to be developed. Assume the failure rate goal of the aggregate is 0.002 failures per hour. Suppose that the mission time is 4 hours. Furthermore, the criticality factors and the total active time of the software CSCIs are:

$c_1 = 4$ $\quad \tau'_1 = 2$ hours
$c_2 = 2$ $\quad \tau'_2 = 1$ hour
$c_3 = 1$ $\quad \tau'_3 = 2$ hours

(Note: In this example, since c_3 has the smallest value, this indicates that the third CSCI of this software aggregate is the most critical.)

Compute the adjustment factor K:

$$K = \frac{c_1 \tau_1 + c_2 \tau_2 + c_3 \tau_3}{T} = \frac{(4)(2) + (2)(1) + (1)(2)}{4} = 3$$

Then, the allocated failure rate goals of the software CSCIs are:

$\lambda_1 = \lambda_s (c_1/K)$
$= 0.002 (4/3) = 0.0027$ failures per hour

$\lambda_2 = \lambda_s (c_2/K)$
$= 0.002 (2/3) = 0.0013$ failures per hour

$\lambda_3 = \lambda_s (c_3/K)$
$= 0.002 (1/3) = 0.00067$ failures per hour

9.6.4 Allocation Based on Complexity Factors

The following technique is used to allocate a failure rate goal to each software CSCI in an aggregate, based on the complexity of the CSCIs. There are several types of complexity as applied to software that are listed in Table 9.6-3.

TABLE 9.6-3: COMPLEXITY PROCEDURES

Complexity Type	Description	When it Can Be Used
McCabe's Complexity	A measure of the branches in logic in a unit of code.	From the start of detailed design on.
Functional Complexity	A measure of the number of cohesive functions performed by the unit.	From the start of detailed design on.
Software Product Research Function Points	A measure of problem, code, and data complexity, inputs, outputs, inquiries, data files and interfaces.	From detailed design on.
Software Product Research Feature Points	A measure of algorithms, inputs, outputs, inquiries, data files and interfaces.	From detailed design on.

During the design phase, an estimated complexity factor using any one of these techniques is available. The greater the complexity, the more effort required to achieve a particular failure rate goal. Thus, CSCIs with higher complexity should be assigned higher failure rate goals.

The complexity measure chosen must be transformed into a measure that is linearly proportional to failure rate. If the complexity factor doubles, for example, the failure rate goal should be twice as high.

Steps:

(1) Determine the failure rate goal of the software aggregate; λ_S
(2) Determine the number of software CSCIs in the aggregate; N
(3) For each $CSCI_i$, i = 1, 2, ..., N, determine its complexity factor; w
(4) Determine the total active time of each $CSCI_i$, i = 1, 2, ..., N; τ_i
(5) Determine the mission time of the aggregates; T
(6) Compute the failure rate adjustment factor K:

$$K = \frac{\sum_{i=1}^{N} w_i \tau_i}{T}$$

(7) Compute the allocated failure rate of the i^{th} CSCI:

$$\lambda_i = \lambda_s (w_i/K)$$

Example:

A software aggregate consisting of 4 software CSCI is to be developed. The failure rate goal of the aggregate is 0.006 failures per hour. The mission time is three hours. Furthermore, the complexity factors and the total active time of the software CSCIs are given as:

$w_1 = 4, \quad \tau_1 = 2$ hours
$w_2 = 2, \quad \tau_2 = 1$ hour
$w_3 = 3, \quad \tau_3 = 3$ hours
$w_4 = 1, \quad \tau_4 = 2$ hours

Compute the failure rate adjustment factor K:

$$K = \frac{\sum_{i=1}^{N} w_i \tau_i}{T} = \frac{(4)(2) + (2)(1) + (3)(3) + (1)(2)}{3} = 7$$

Then, the failure rate goal of each software CSCIs is:

$\lambda_1 = \lambda_s (w_1/K)$
= 0.006 (4/7) = 0.0034 failures per hour

$\lambda_2 = \lambda_s (w_2/K)$
= 0.006 (2/7) = 0.0017 failures per hour

$\lambda_3 = \lambda_s (w_3/K)$
= 0.006 (3/7) = 0.00026 failures per hour

$\lambda_4 = \lambda_s (w_4/K)$
= 0.006 (1/7) = 0.0009 failures per hour

9.7 Software Testing

Most software experts recommend that an independent organization test a software system. One option is to contract with an outside organization for the testing. If this is not possible, the testing organization should be managerially separate from the design and development groups assigned to the project.

This recommendation is based more on observations of human nature than on substantiated fact. Effective testing groups need to have somewhat of a "destructive" view of a system, so that they can flush out errors and "break" the system. The design and development groups who have built the software system have a "constructive" view, and may therefore find it too difficult to develop the frame of mind required for testing.

9.7.1 Module Testing

Module testing (also called unit or component testing) is the testing of one individual component (that is, one program module, one functional unit, or one subroutine). The objective of module testing is to determine if the module functions according to its specifications.

Module testing is usually conducted by the programmer of the module being tested. It is closely tied to the programmer's development of the code and often becomes an iterative process of testing a component, finding a problem, debugging (finding the reason for the problem in the code), fixing the problem, and then testing again. Module testing is therefore often considered part of the implementation rather than part of the testing phase. Module testing should nevertheless be recognized as a separate function, and should be disciplined. The tester must develop a test plan for the component and must document test cases and procedures. Too often, this discipline is overlooked and testing of individual components becomes "ad hoc" testing with no records about the actual cases, the procedures, or the results.

White box testing is frequently used during module testing. White box testing means that the tester is familiar with the internal logic of the component and develops test cases accordingly.

Code coverage (how much of the code is covered by the testing) and logic path coverage (how many of the logical paths in the code are tested) are two primary considerations when developing test cases for module testing.

9.7.2 Integration Testing

After module testing, the next step in the software testing phase is integration testing. This activity involves combining components in an orderly progression until the entire system has been built. The emphasis of integration testing is on the interaction of the different components and the interfaces between them.

Most often, the programming group performs software integration testing. As with module testing, integration testing is very closely linked to the programming activity since the tester needs to know details of the function of each component to develop a good integration test plan.

Integration Test Techniques. An important decision when planning for integration testing is determining the procedure to be used for combining all the individual modules. There are two basic approaches for doing this: non-incremental testing and incremental testing.

In non-incremental integration testing, all the software components (assuming they have each been individually module tested) are combined at once and then testing begins. Since all modules are combined at once, a failure could be in any one of the numerous interfaces that have been introduced.

The recommended approach for the integration of system components is planned incremental testing. With this method, one component is completely module tested and debugged. Another component is then added to the first and the combination is tested and debugged. This pattern of adding one new component at a time is repeated until all components have been added to the test and the system is completely integrated.

Incremental testing requires another decision about the order in which the components will be added to the test. There are no clear-cut rules for doing this. Testers must base a decision on their knowledge of what makes the most sense for their system, considering logic and use of resources. There are two basic strategies: top-down or bottom-up as shown in Figure 9.7-1.

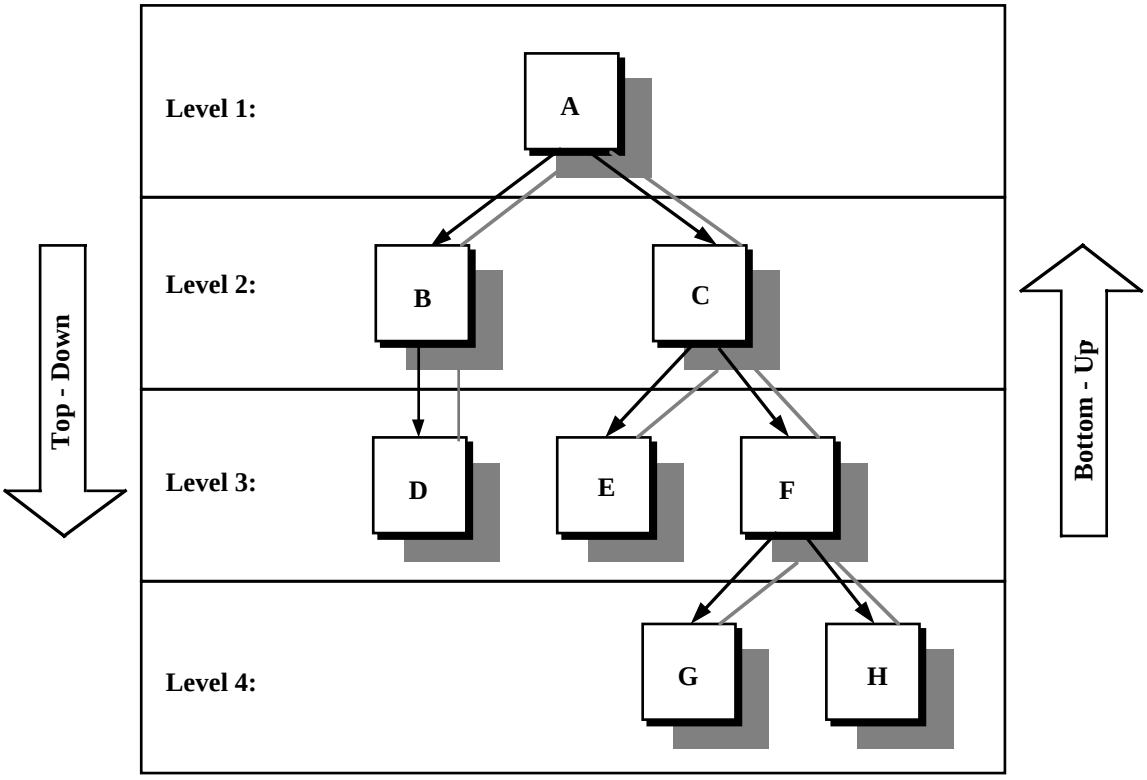


FIGURE 9.7-1: STRUCTURAL REPRESENTATION OF A SOFTWARE SYSTEM

A tester using top-down integration testing on this system begins by module testing and debugging the A component. The next step is to add a new component to the test. In this case, either B or C is added. If B was chosen and tested, either C or D could be the next choice. Some testers prefer to follow one path to completion, while others prefer to complete all the modules on the same level before proceeding to a lower level of the hierarchy.

Bottom-up integration testing reverses top-down testing. With this approach, a tester simply starts at the bottom-most level of the hierarchy and works up. As shown in Figure 9-16, a tester might start by module testing component G. With bottom-up testing, all the components at the bottom of the hierarchy are usually module tested first and then testing proceeds in turn to each of their calling components. The primary rule in bottom-up testing is that a component should not be chosen to be the next one added to the test unless all of the components that it calls have already been tested.

9.7.3 System Testing

System Testing Techniques. System testing is often referred to as "testing the whole system." Translated literally, that could mean that every input or output condition in the software needs to be tested for every possible logical path through the code. Even in a small system this task could become quite lengthy. In a large, complex system, it would be prohibitively time-consuming and expensive.

The system test organization must develop a strategy for testing a particular system and determine the amount of test coverage required. There is no cookbook for doing so. In a small noncritical system, a very low degree of test coverage may be acceptable. High coverage is needed in a critical software system involving human life. The testers must decide the best plan based on system characteristics, the environment in which the software system will operate, and the testers' experience.

In general, software system testing is done using black box testing. The tester, viewing the system as a black box, is not concerned with the internals, but rather is interested in finding if and when the system does not behave according to its requirements.

One technique often used for identifying specific test cases is called equivalence partitioning. In this method, an equivalence class is identified so that one test case covers a number of other possible test cases.

Boundary analysis is another technique used in which testing is performed on all the boundary conditions. This method tests the upper and lower boundaries of the program. In addition, it is usually wise to test around the boundaries.

A third technique that should always be applied to the testing of a program is called error guessing. With this method, testers use their intuition and experience to develop specific test cases. A good system tester is usually very effective at doing this.

9.7.4 General Methodology for Software Failure Data Analysis

A step-by-step procedure for software failure data analysis is shown in Figure 9.7-2 and described below:

Step 1: Study the failure data

The models previously described assume that the failure data represent the data collected after the system has been integrated and the number of failures per unit time is statistically decreasing. If, however, this is not the case, these models may not yield satisfactory results. Furthermore, adequate amount of data must be available to get a satisfactory model. A rule of thumb would be to have at least thirty data points.

Step 2: Obtain estimates of parameters of the model

Different methods are generally required depending upon the type of available data. The most commonly used ones are the least squares and maximum likelihood methods.

Step 3: Obtain the fitted model

The fitted model is obtained by first substituting the estimated values of the parameters in the postulated model. At this stage, we have a fitted model based on the available failure data.

Step 4: Perform goodness-of-fit test

Before proceeding further, it is advisable to conduct the Kolmogorov-Smirnov goodness-of-fit test or some other suitable test to check the model fit.

If the model fits, we can move ahead. However, if the model does not fit, we have to collect additional data or seek a better, more appropriate model. There is no easy answer to either how much data to collect or how to look for a better model. Decisions on these issues are very much problem dependent.

Step 5: Computer confidence regions

It is generally desirable to obtain 80%, 90%, 95%, and 99% joint confidence regions for the parameters of the model to assess the uncertainty associated with their estimation.

Step 6: Obtain performance measure

At this stage, we can compute various quantitative measures to assess the performance of the software system. Confidence bounds can also be obtained for these measures to evaluate the degree of uncertainty in the computed values.

9.8 Software Analyses

Two types of analyses will be discussed in this section; the failure modes and effects analysis (FMEA) and the fault tree analysis (FTA). The objective of both analyses is to determine what the system or product software may do or not do that is not desirable. This is opposite of most analyses which attempt to show that the product performs the intended functions. Safety criticality is one area for detailed software analysis.

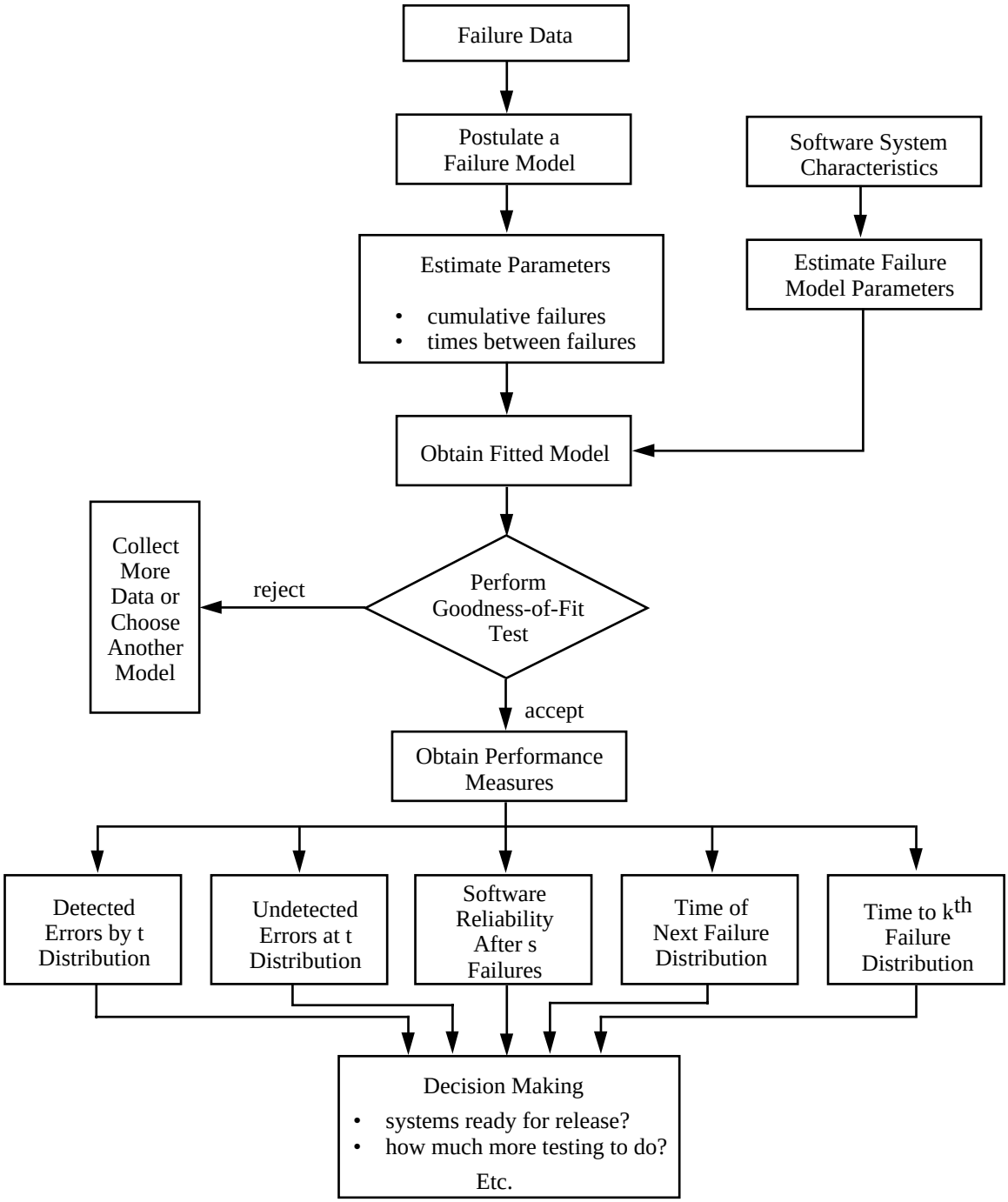


FIGURE 9.7-2: FLOWCHART FOR SOFTWARE FAILURE DATA ANALYSIS AND DECISION-MAKING

9.8.1 Failure Modes

The definition of what constitutes a failure in software is often open to debate. When a program "*crashes*", it has obviously failed due to an error, either in design, coding, testing, or exception handling.

Software sometimes fails to perform as desired. These failures may be due to errors, ambiguities, oversights or misinterpretation of the specification that the software is supposed to satisfy, carelessness or incompetence in writing code, inadequate testing, incorrect or unexpected usage of the software or other unforeseen problems. (Ref. [25]).

However, the software may not crash and still fail. This can be due to a number of criteria which are not always well defined before development. Speed of execution, accuracy of the calculations and other criteria can all be significant factors when identifying lack of successful software operation. In addition, each of these criteria has a specific level of importance and is assessed on a specific scale.

When first using or evaluating a software program, a significant amount of time can be spent determining compliance with specification or applicability of an application. Depending on the person doing the evaluation, or the evaluation process scope and design, the evaluation may or may not fully exercise the program or accurately identify functionality that is unacceptable.

Hardware/software interface problems can also occur, including failures in software due to hardware or communications environment modifications. Software errors can be introduced during software functional upgrades or during either scheduled or unscheduled maintenance.

Software failures are usually considered relative to the application type and the severity of failure as evaluated by the specific end user. Consider the following two examples. One software program contains complex graphical user interfaces that map exactly to the customer's layout requirements; but this program crashes whenever a specific sequence of user inputs and events occurs. Another software program has layout flaws but it does not fail for any sequence of user triggered events. Which program is more reliable?

(1) Is an application reliable if it meets all specified requirements? Then the first is better.

(2) If failure is defined as any crash, then the second is more reliable; in fact, some would say it is perfectly reliable because it does not crash.

9.8.2 Failure Effects

When software fails, a dramatic effect, such as a plane crash, can also be observed. Often, however, the effect of a software failure is not immediately seen or may only cause inconvenience. A supermarket checkout system which incorrectly prices selected items may never be noticed, but a failure has still occurred. An Automatic Teller Machine (ATM) which

does not allow user access is a nuisance which results in disgruntled customers. Both of these may be the result of catastrophic software failures, but, in reference to endangering human life, both are minor system failures.

These examples illustrate that is it important to distinguish between the software failure relative to the software's functioning as compared to the software failure relative to the total system's functioning. In the supermarket example, the software may have failed but the checkout continued while in the ATM example, the system did not operate.

9.8.3 Failure Criticality

Both hardware and software fall into two general categories based on the function performed: *mission critical* and *non-mission critical*. *Mission critical* encompasses all failures that are life threatening as well as failures that have catastrophic consequences to society. Table 9.8-1 identifies hardware failure severity levels with respect to both mission and operator. In hardware reliability improvement, usually only catastrophic and critical levels of severity are addressed.

TABLE 9.8-1: HARDWARE FAILURE SEVERITY LEVELS (REF. [26])

Term	Definition
Catastrophic	A failure which may cause death or system loss (i.e., aircraft, tank, missile, ship, etc.).
Critical	A failure which may cause severe injury, major property damage, or major system damage which will result in mission loss.
Marginal	A failure which may cause minor injury, minor property damage, or minor system damage which will result in delay or loss of availability or mission degradation.
Minor (Negligible)	A failure not serious enough to cause injury, property damage, or system damage, but which will result in unscheduled maintenance or repair.

No similar set of criticality classifications has been adopted by the entire software community. Putnam and Myers have defined four classes of software defect severity and identify the corresponding user response as shown in Table 9.8-2. It is interesting to note that this classification is not with respect to operator or mission but views the software as an entity in itself. No application reference is included in the descriptions. Another interesting contrast is that any level of software defect can cause a catastrophic system failure. If the software crashes ("*Critical*"), mis-computes ("*Serious*"), provides a partly correct answer ("*Moderate*") or mis-displays the answer on the screen ("*Cosmetic*"), the resultant failure may be catastrophic, resulting in system and/or operator loss.

TABLE 9.8-2: SOFTWARE FAILURE SEVERITY LEVELS (REF. [5])

Severity	Description	User Response
Critical	Prevents further execution; nonrecoverable.	Must be fixed before program is used again.
Serious	Subsequent answers grossly wrong or performance substantially degraded.	User could continue operating only if allowance is made for the poor results the defect is causing. Should be fixed soon.
Moderate	Execution continues, but behavior only partially correct.	Should be fixed in this release.
Cosmetic	Tolerable or deferrable, such as errors in format of displays or printouts.	Should be fixed for appearance reasons, but fix may be delayed until convenient.

9.8.4 Fault Tree Analysis

Fault tree analysis is performed on software to determine the areas in the product which could cause a potential failure and to determine the risk and severity of any such potential failure. The timing of this analysis is important and should start during the design phase to identify top-level hazards. The analysis can continue through code development and testing to identify paths for testing and verify that safety related hazards will not occur.

The steps for performing a software fault tree are:

(1) Determine failure modes for software starting from top level product and working downward.

(2) Make these failure modes the top nodes of the fault tree. Assume that these failure modes have already occurred and refer to them as events.

(3) When tree completed for top level failure modes, determine risk and severity for each of the bottom nodes on the tree.

Risk (Ref. [27])

1	Remote possibility of happening
2-3	Low probability with similar designs
4-6	Moderate probability with similar designs
7-9	Frequent probability with similar designs
10-	High probability with similar design

Severity (Ref. [27])

1-2	Probably not detected by customer
3-5	Result in slight customer annoyance
6-7	Results in customer dissatisfaction
8-9	Results in high customer dissatisfaction
10-	Results in major customer dissatisfaction, loss of system operation, or non-compliance with government regulations

(4) Using design flows and charts, determine how the failure modes identified in step 1 can occur. Assume that the failure mode has already occurred and identify what causes it. Bottom nodes of tree will be more specific failure modes.

(5) When the tree is completed for the next level of design, identify failure modes associated with this level and identify risk and severity.

(6) Repeat steps 4 and 5 until a satisfactory level of abstraction has been reached (normally determined by customer).

(7) The tree is pruned when risk and severity are insignificant or when the lowest level of abstraction is reached. The tree is expanded when risk and probability are significant.

9.8.5 Failure Modes and Effects Analysis

In contrast to the fault tree top down development, the failure modes and effects analysis is a bottom up approach. That is, a failure mode is selected in a lower level unit and the failure effect through the system is evaluated. The units that will be affected and the probability and criticality of the failure mode is determined from the failure rates, operating time, and criticality ranking.

The steps for applying failure modes and effects analysis to software are:

(1) Determine product level failure modes using step 1 of the Fault Tree Analysis section.
(2) Using a software failure mode chart, work through top level of chart using the top level failure modes and fill in the form. One unit may have several failure modes or no failure modes.
(3) Repeat steps 1 and 2 for the next level of design until lowest level is reached.

Table 9.8-3 lists the categories that must be addressed when performing a complete failure mode and criticality analyses.

An example of a software failure modes and effects analysis is shown in Figure 9.8-1. Each function failure mode and end effect are described.

TABLE 9.8-3: SOFTWARE FAILURE MODES AND CRITICALITY ANALYSIS CATEGORIES

Software FMECA Categories

(1) Unit - Name of software unit at CSCI, CSC or unit level
(2) Function - General function performed by unit
(3) Failure mode - the associated failure mode
(4) The probable cause of the failure in software terms
(5) The effect on the unit, the next level and the top level. Define these effects in terms of processing, output, etc.
(6) Interrupted? If service/mission would be interrupted by this failure mode state so.
(7) Crit - Criticality I - catastrophic, II - critical, III - moderate, IV - negligible
(8) Predictability - If there is some predictability before the failure occurs state so. Normally software failure have no predictability so this will probably always be no
(9) Action - the type of corrective action required. This will either be restart if the problem can be circumvented, or remote corrective action if it can only be fixed in an engineering environment.

No.	Unit	Function	Failure Mode	Probable Cause	Effect On			Interrupt ?	Crit	Action
					Unit	Sub	System			
1	Output	Outputs file into	Output is incorrect	Inputs are invalid and not detected	n/a	none	mission degraded	no	II	lab repair
2	Output	Outputs file into	Output is incorrect	Inputs are correct but not stored properly	n/a	none	mission degraded	no	II	lab repair
3	Output	Outputs file into	Output is incorrect	Values are not computed to spec	n/a	none	mission degraded	no	II	lab repair

FIGURE 9.8-1: EXAMPLE OF SOFTWARE FMECA

9.9 References

1. Coppola, Anthony, "*Reliability Engineering of Electronic Equipment - A Historical Perspective*," IEEE Transactions on Reliability, April 1984, pp. 29-35.

2. Lakey, P.B., and A.M. Neufelder, "*System and Software Reliability Assurance,"* RL-TR-97-XX, Rome Laboratory, 1997.

3. Jones, Capers, "Assessment and Control of Software Risks," Yourdon, 1994.

4. Dunn, Robert, "*Software Defect Removal,"* McGraw-Hill, 1984.

5. Putnam, Lawrence H., and Myers, Ware, "Measures for Excellence: Reliable Software on Time, Within Budget," Prentice-Hill, 1992.

6. Boehm, Barry W., "*Software Engineering Economics*," Prentice-Hall, 1981.

7. Rook, Paul, Editor, "*Software Reliability Handbook*," Elsevier, 1990.

8. Dyer, Michael, "The Cleanroom Approach to Quality Software Development," Wiley, 1992.

9. Mills, Harlan D., "*Cleanroom Software Engineering,"* Center for Software Engineering, 1994.

10. Linger, Richard C., "Cleanroom Process Model," IEEE Software, March 1994, pp. 50-58.

11. Software Technology Support Center, "Guidelines for Successful Acquisition and Management of Software Intensive Systems," Dept. of the Air Force, February 1995.

12. Musa, John, A. Iannino, K. Okumoto, "*Software Reliability: Measurement, Prediction, Application*," McGraw-Hill, 1987.

13. Fairley, .R.E., "*Software Engineering Concepts*," McGraw-Hill, 1985.

14. Booch, Grady, "*Object-Oriented Development,"* IEEE Transactions of Software (February), 1986.

15. Meyer, Bertrand, "*Object-Oriented Construction,"* Prentice-Hall, 1988.

16. McCall, J.A., W. Randell, and J. Dunham, "*Software Reliability, Measurement, and Testing,"* Rome Laboratory, RL-TR-92-52, 1992.

17. Friedman, M.A., P.K. Tran and P.L. Goddard, *"Reliability Techniques for Combined* Hardware and Software Systems," RL-TR-92-15, 1992.

18. Lloyd D.K. and M. Lipow, *"Reliability: Management, Methods, and Mathematics,"* Second Edition, American Society for Quality Control, 1977.

19. Farr, W.H., *"A Survey of Software Reliability Modeling and Estimation,"* Naval Surface Weapons Center, NSWC-TR-82-171, 1983.

20. Shooman, Martin L., *"Software Engineering,"* McGraw Hill, 1983.

21. IEEE STD 982.1, "IEEE Standard Dictionary of Measures to Produce Reliable Software," 1989.

22. IEEE STD 982.2, "Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software," 1989.

23. Neufelder, A., *"Ensuring Software Reliability,"* Marcel Dekker, 1993.

24. Mahar, David, *"Fault Tree Analysis,"* Reliability Analysis Center (RAC), 1990.

25. Keiller, Peter A. and Douglas R. Miller, *"On the Use and the Performance of Software Reliability Growth Models,"* Software Reliability and Safety, Elsevier, 1991.

26. MIL-STD-1629A, "Procedure for Performing Failure Mode, Effects and Criticality Analysis," Department of Defense, 1980.

27. Texas Instruments, Inc., "Components Sector Training and Organizational Effectiveness, Failure Modes and Effect Analysis," 1993.

10.0 SYSTEMS RELIABILITY ENGINEERING

10.1 Introduction

The material presented in the previous sections of this handbook in a sense set the stage for this section. This section combines the R&M theory and engineering practices previously presented into a cohesive design methodology which can be applied at the system level to optimize system "worth" for minimum life cycle costs.

The "worth" of a particular equipment/system is determined primarily by the effectiveness with which it does its job - its "operational" effectiveness. An acceptable level of effectiveness is required for every operational system.

In the final analysis, the effectiveness of a system can only be really measured when the system is performing its mission in the actual (or accurately simulated) environment for which it was designed. Of critical importance, however, is how system effectiveness can be considered while system design concepts are developed, how it can be ensured during design, and how it can be evaluated during test. Thus, most system effectiveness methodologies address these issues more than measuring system effectiveness after the system is fielded.

Table 10.1-1 represents the system effectiveness concept and the parameters that have been traditionally used (with minor variations) for system effectiveness analysis.

TABLE 10.1-1: CONCEPT OF SYSTEM EFFECTIVENESS

	System Effectiveness is the Net Result of		
	Availability	Dependability	Capability
Measures:	System condition at start of mission	System condition during performance of mission	Results of mission
Determined by:	Reliability Maintainability Human Factors Logistics	Repairability Safety Survivability Vulnerability	Range Accuracy Power Lethality etc.

As can be seen from the table, availability (how often), dependability (how long), and performance capability (how well) are the primary measures of system effectiveness:

(1) **Availability** is a measure of the degree to which an item is in the operable and committable state at the start of the mission, when the mission is called for at an unknown (random) time.

(2) **Dependability** is a measure of the degree to which an item is operable and capable of performing its required function at any (random) time during a specified mission profile, given item availability at the start of the mission. (This definition is different than the definition of dependability as it appears in International Electrotechnical Commission documents.)

(3) **Capability** is a measure of the ability of an item to achieve mission objectives, given the conditions during the mission.

System effectiveness assessment fundamentally answers three basic questions:

(1) Is the system working at the start of the mission?

(2) If the system is working at the start of the mission, will it continue to work during the mission?

(3) If the system worked throughout the mission, will it achieve mission success?

R&M are important contributions to system effectiveness since they are significant factors in consideration of the availability and dependability parameters. However, in the total system design context, as shown in Table 10.1-1, they must be integrated with other system parameters such as performance, safety, human factors, survivability/vulnerability, logistics, etc., to arrive at the optimum system configuration.

Just about all of the system effectiveness methodologies which have been developed and/or proposed in the past 20 years are concerned with this fundamental question of combining the previously mentioned parameters to achieve optimum system design. In Section 10.2, some of the more significant system effectiveness concepts and methodologies are discussed and compared.

10.1.1 Commercial-Off-The-Shelf (COTS) and Nondevelopmental Item (NDI) Considerations

Under the current military acquisition reform initiatives, the Department of Defense is advocating the use of Commercial-Off-The-Shelf (COTS) and Nondevelopmental Items (NDI) in the products it acquires for military applications. Commercial industry has long used NDI in building new products. NDI is any previously developed item used exclusively for government purposes by “federal agency, a state or local government or a foreign government with which the

US has mutual defense cooperation agreement."[1] COTS are items available in a domestic or foreign commercial marketplace. The increased emphasis on commercial products and practices has occurred for a number of reasons. First, the decrease in military spending over the last decade has resulted in an erosion in the industrial base that existed to support development of weapon systems. Second, while technology was driven primarily by the DoD in the past, this is no longer the case. Third, many technologies (e.g., electronics, information, communications) are advancing at such a rapid pace that the government can no longer afford an acquisition process that has historically required at least a 2-3 year cycle to develop, test, and field a system.

The objective of using COTS/NDI is to reduce the development time and risk associated with a new product by reducing or eliminating new design and development, thereby capitalizing on proven designs. Whether it is the government or a private commercial company, using COTS/NDI can potentially reduce costs, risks, and acquisition time. However, some compromises in the required functional performance (including reliability) of the product may be necessary, and other issues, such as logistics support, must also be considered. The decision to use COTS/NDI must be based on a thorough evaluation of its ability to perform the required function in the intended environment and to be operated and supported over the planned life of the product.

A product that is new in every aspect of its design carries with it cost, schedule, and performance risks. These risks are usually high for such a product because of all the unknowns surrounding a totally new design. A product development involving a completely new design is considered revolutionary in nature.

In contrast to a completely new design (revolutionary approach), using a proven product or incorporating proven components and subsystems in a new product is an evolutionary approach. Using COTS/NDI is a way to follow a pattern of new product development in which new design is minimized or eliminated. Some types of NDI include:

- Items available from a domestic or foreign commercial marketplace
- Items already developed and in use by the U.S. government
- Items already developed by foreign governments

COTS/NDI items may constitute the entire product (e.g., a desktop computer) or they may be components or subsystems within the product (e.g., displays, power supplies, etc., used within a control system). The advantages and disadvantages of using COTS/NDI are summarized in Table 10.1-2.

The use of commercial items in military systems is no longer a question of "yes or no" but a question of "to what degree." A pictorial presentation of the commercial/ NDI decision process is shown in Figure 10.1-1 taken from SD-2. The R&M activities needed for COTS/NDI are different than for new development items, as shown in Table 10.1-3. These considerations are

[1] SD-2, Buying Commercial and Nondevelopment Item: A Handbook, Office of the Assistant Secretary of Defense for Production and Logistics, April 1996.

discussed in more detail in the following paragraphs.

For new development programs, the customer imposes reliability requirements in the system specification and development specifications. (In addition, prior to Acquisition Reform, the customer stipulated in the statement of work which tasks the contractor would conduct as part of the reliability program and how (by imposing standards) the tasks were to be conducted).

With commercial items and NDI, the basic product is already designed and its reliability established. Consequently, the reliability assessment should be an operational assessment of the military application in the expected military environments. Since the basic design of a commercial or nondevelopmental item cannot be controlled by the buyer, the objective is to determine whether well-established and sound reliability practices were applied during the item's development.

When considering the use of COTS/NDI equipment, much work needs to be done up front in terms of market research and development of minimum requirements. This means that procurement offices must work closely with the end user to define the minimum acceptable performance specifications for R&M. Market research then needs to be performed to see what COTS/NDI equipment exists that has the potential of meeting defined requirements at an affordable price.

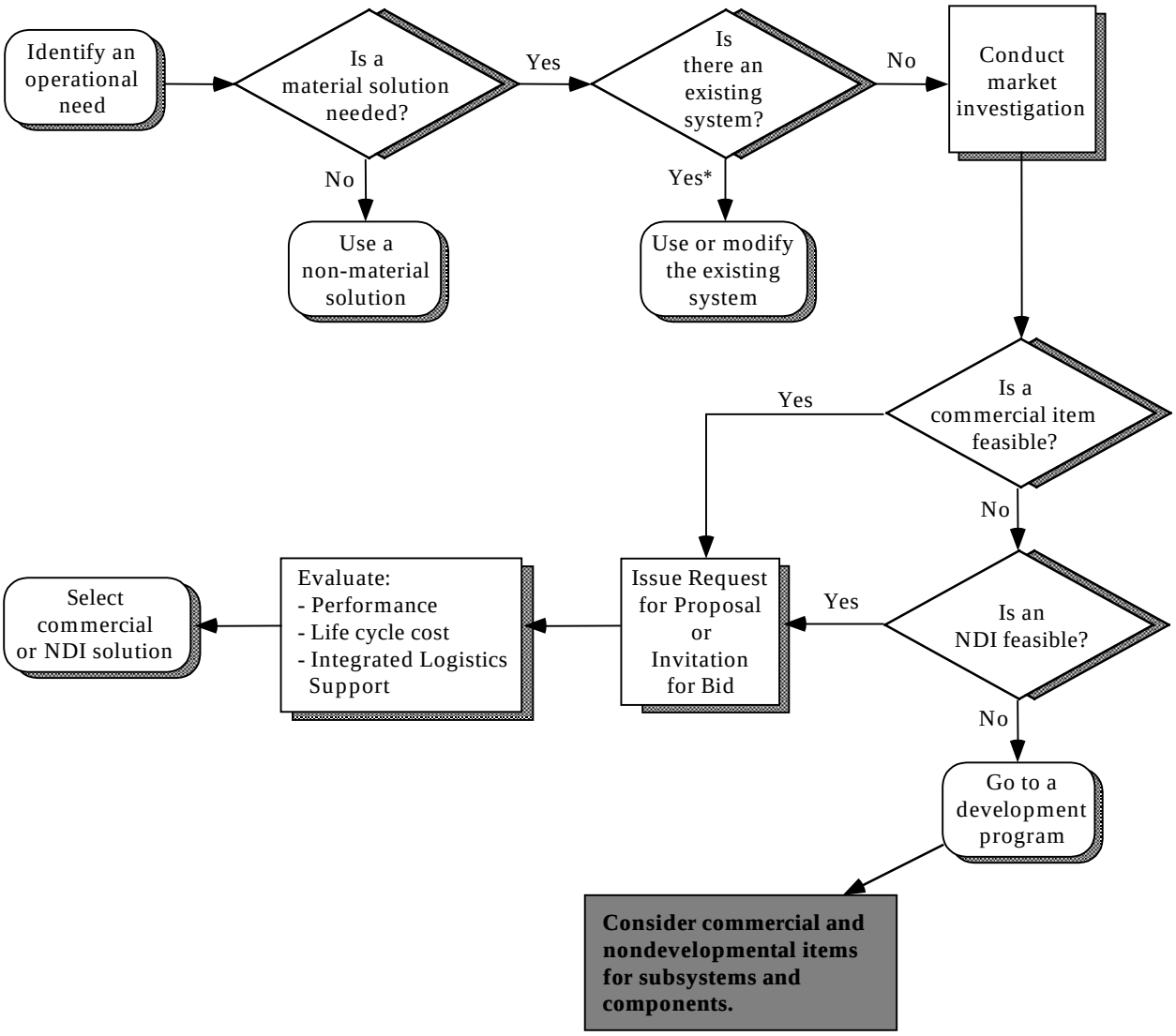
The challenge for market research is obtaining R&M data on COTS/NDI equipment. COTS vendors may not have the kinds of data that exist in military R&M data collection systems. (Text continues after Tables 10.1-1 and 10.1-2 and Figure 10.1-1).

TABLE 10.1-2: ADVANTAGES AND DISADVANTAGES OF COTS/NDI

AREA OF COMPARISON	ADVANTAGES	DISADVANTAGES
Technical, Schedule, and Financial Risk	Decreased technical, financial, and schedule risks due to less new design of components and subsystems. Ideally no research and development costs are incurred.	When NDI items are used as the components and subsystems of a product, integration of those items into the product can be difficult, expensive, and time-consuming.
Performance	There is increased confidence due to established product performance and the use of proven components and subsystems.	Performance trade-offs may be needed to gain the advantages of NDI. Integration may be difficult.
Environmental Suitability	In similar applications, proven ability to operate under environmental conditions.	In new applications, may require modifications external or internal to the equipment to operate.
Leverage	Ability to capitalize on economies of scale, state-of-the-art technology, and products with established quality.	There may not be a perfect match between requirements and available products.
Responsiveness	Quick response to an operational need is possible because new development is eliminated or minimized.	Integration problems may reduce the time saved.
Manufacturing	If already in production, processes are probably established and proven.	Configuration or process may be changed with no advance notice.
Resupply	There is no need for (large) inventory of spares because they can be ordered from supplier.	The long-term availability of the item(s), particularly COTS, may be questionable.
Logistics Support	No organic support may be required (probably not possible). Repair procedures and rates are established.	Supplier support or innovative integrated logistics support strategies may be needed to support the product.

TABLE 10.1-3: R&M ACTIVITIES FOR NEW DEVELOPMENT ITEMS AND FOR COTS

R&M ACTIVITY	TYPE OF ITEM	
	NEW DEVELOPMENT	**COTS/NDI**
Determine Feasibility	Develop requirements based on user needs and technology being used. Estimate achievable level of R&M.	Limited to verifying manufacturer claims.
Understand the Design	Perform FMEA, FTA, and other analyses for entire design. Conduct design reviews. Develop derating criteria. Conduct development testing.	Limited to integration and any modifications.
Parts Selection	Analyze design to determine correct parts application for robust design. Identify needed screening.	None.
Validate the Design	Conduct extensive development testing that addresses all aspects of the design. Identify design deficiencies and take corrective action. Establish achieved levels of R&M.	Limited to what is needed to verify manufacturer claims and to validate integration or required modifications based on the intended environment.
Manufacturing	Design manufacturing processes to retain inherent R&M. Implement statistical process control and develop good supplier relationships.	None if the item is already in production. Otherwise, design the manufacturing process to retain the inherent design characteristics.



* In preparation for the market investigation establish objectives and thresholds for cost, schedule, and performance based on the users' operational and readiness requirements.

FIGURE 10.1-1: THE COMMERCIAL/NDI DECISION PROCESS

If design documentation is available, specific R&M tasks, such as prediction and failure modes and effects analysis, may be part of the COTS/NDI evaluation process. Because the prime military contractor is not likely to be the COTS/NDI vendor in this case, both the government and the prime will need to perform the evaluation i.e., a cooperative effort should exist between the two parties.

The amount of testing required to verify that a commercial item or NDI meets the operational requirement is governed by whether the item will be used in the environment for which it was designed and by operators with skills equal to the operators for which it was designed. What may be needed is to require the supplier to furnish operational and environmental characterization data and the results of testing to substantiate reliability and maintainability claims. Also, it may be necessary to require the supplier provide some evidence that the manufacturing processes do not compromise the designed-in reliability and maintainability characteristics. This evidence may include the results of sampling tests, control charts showing that critical processes are in control with a high process capability, and so forth.

10.1.2 COTS/NDI as the End Product

When purchasing COTS/NDI as the total product, the best course of action may be to require only data that substantiates R&M performance claims and to emphasize the role of the manufacturing processes (for NDI not yet in production) in determining the reliability and maintainability of the product. In some cases, even that data may not be needed if either the customer has already determined (through its own testing of samples, for example) that the product has the requisite performance, or if use or independent testing of the product in commercial applications has shown the product's performance to be satisfactory (for example, a personal computer in an office environment). In any case, imposing specific R&M tasks on manufacturers of COTS/NDI, even if they were willing to bid on such a procurement, is usually counterproductive and expensive.

The advantage of using COTS/NDI is that the development is complete (with only minor exceptions); the supplier has already done (or omitted) whatever might have been done to design a reliable and maintainable product. What may be need is to require the supplier to furnish operational and environmental characterization data and the results of testing to substantiate reliability and maintainability claims. Also, it may be necessary to require the supplier provide some evidence that the manufacturing processes do not compromise the designed-in reliability and maintainability characteristics. This evidence may include the results of sampling tests, control charts showing that critical processes are in control with a high process capability, and so forth.

10.1.3 COTS/NDI Integrated with Other Items

When COTS/NDI is being integrated with other items, either new development or other COTS/NDI, the same attention and level of effort that is characteristic of a new development must be given to the integration. R&M and other performance characteristics may be seriously

affected by the integration due to feedback, interference and other interactions. The integration may require interface devices, which themselves may present new R&M problems. One would expect a supplier to perform Failure Modes and Effects Analysis (FMEA), Fault Tree Analysis (FTA), and other analyses to ensure that the integration does not compromise the R&M performance of any of the items being integrated and that the resulting product meets the required levels of R&M performance.

10.1.4 Related COTS/NDI Issues

Three of the most important issues associated with using COTS/NDI are the logistics support concept, availability of parts, and performance in the intended military environment. Other issues include configuration management of the COTS/NDI (important if the customer plans to support the product organically), and the availability or development of documentation to support operations and organic maintenance.

10.2 System Effectiveness Concepts

The three generally recognized components of system effectiveness previously defined (availability, dependability, capability) will be used as the basis for description and comparison of the concepts and formulations of system effectiveness. It should be recognized that all of these effectiveness components must be derived from an analysis of the operational needs and mission requirements of the system, since it is only in relation to needs and missions that these basic components can be meaningfully established.

Many semantic difficulties arise when discussing systems effectiveness and its components. These difficulties result from the fact that some people use the same words to mean different things or different words to mean the same things.

Definitions of many of the terms used in the following paragraphs were provided in Section 3 and will not be repeated here.

10.2.1 The ARINC Concept of System Effectiveness (Ref. [1])

One of the early attempts to develop concepts of system effectiveness was delineated by ARINC (Aeronautical Radio Inc.) in its book “Reliability Engineering.” It contains some of the earliest published concepts of systems effectiveness and represents one of the clearest presentations of these concepts from which many of the subsequent descriptions have been derived. The definition of systems effectiveness applied in this early work is: “Systems effectiveness is the probability that the system can successfully meet an operational demand within a given time when operated under specified conditions.” This definition includes the concepts that system effectiveness

(1) Can be measured as a **probability**

(2) Is related to operation **performance**

(3) Is a function of **time**

(4) Is a function of the **environment** or conditions under which it is used

(5) May vary with the **mission** to be performed

Although it is not essential to describe system effectiveness in terms of probability as opposed to other quantitative measures, it has often been found convenient to do so. The ARINC model may be expressed such that system effectiveness probability, P_{SE}, is the product of three probabilities as follows:

$$P_{SE} = P_{OR} \cdot P_{MR} \cdot P_{DA} \qquad (10.1)$$

where:

P_{OR} = operational readiness probability
P_{MR} = mission reliability probability
P_{DA} = design adequacy probability

This equation states that the effectiveness of the system is the product of three probabilities: (1) the probability that the system is operating satisfactorily or is ready to be placed in operation when needed; (2) the probability that the system will continue to operate satisfactorily for the period of time required for the mission; and (3) the probability that the system will successfully accomplish its mission, given that it is operating within design limits.

10.2.2 The Air Force (WSEIAC) Concept (Ref. [2])

A later definition of system effectiveness resulted from the work of the Weapon System Effectiveness Industry Advisory Committee (WSEIAC) established in late 1963 by the Air Force System Command. The WSEIAC definition of system effectiveness is: "System effectiveness is a measure of the extent to which a system may be expected to achieve a set of specific mission requirements and is a function of availability, dependability, and capability." The definition may be expressed as:

$$SE = ADC \qquad (10.2)$$

where:

A = availability
D = dependability
C = capability

See definitions in Section 10.1.

These are usually expressed as probabilities as follows:

(1) "A" is the vector array of various state probabilities of the system at the beginning of the mission.

(2) "D" is a matrix of conditional probabilities over a time interval, conditional on the effective state of the mission during the previous time interval.

(3) "C" is also a delineal probability matrix representing the performance spectrum of the system, given the mission and system conditions, that is, the expected figures of merit for the system.

Basically, the model is a product of three matrices:

- Availability row vector A
- Dependability matrix D
- Capability matrix C

In the most general case, assume that a system can be in different states and at any given point in time is in either one or the other of the states. The **availability row vector** is then

$$\mathbf{A} = (a_1 , a_2 , a_3 , \ldots , a_i , \ldots a_n) \qquad (10.3)$$

where a_i is the probability that the system is in State i at a random mission beginning time. Since the system can be in only one of the n states and n is the number of all possible states it can be in (including the down states in which the system cannot start a mission), the sum of all the probabilities a_i in the row vector must be unity, i.e.,

$$\sum_{i=1}^{n} a_i = 1 \qquad (10.4)$$

The **dependability matrix** D is defined as a square n • n matrix

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ d_{n1} & d_{n2} & d_{n3} & \cdots & d_{nn} \end{bmatrix} \qquad (10.5)$$

where the meaning of the element d_{ij} is defined as the expected fraction of mission time during which the system will be in State j if it were in State i at the beginning of the mission. If system output is not continuous during the mission but is required only at a specific point in the mission (such as over the target area), d_{ij} is defined as the probability that the system will be in State j at the time when output is required if it were in State i at mission start.

When no repairs are possible or permissible during a mission, the system upon failure or partial failure cannot be restored to its original state during the mission and can at best remain in the State i in which it started the mission or will degrade into lower states or fail completely. In the case of no repairs during the mission, some of the matrix elements become zero. If we define State 1 as the highest state (i.e., everything works perfectly) and n the lowest state (i.e., complete failure), the dependability matrix becomes triangular with all entries below the diagonal being zeros.

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ 0 & d_{22} & d_{23} & \cdots & d_{2n} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & d_{nn} \end{bmatrix} \qquad (10.6)$$

If the matrix is properly formulated the sum of the entries in each row must equal unity. For example, for the first row we must have

$$d_{11} + d_{12} + \ldots + d_{1n} = 1 \qquad (10.7)$$

and the same must apply to each subsequent row. This provides a good check when formulating a dependability matrix.

The **capability matrix**, C, describes system performance or capability to perform while in any of

the n possible system states. If only a single measure of system effectiveness is of importance or of interest, C will be a one column matrix with n elements, such as

$$\mathbf{C} = \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} \tag{10.8}$$

where c_j represents system performance when the system is in State j.

System effectiveness, SE, in the WSEIAC model is then defined as

$$SE = \begin{bmatrix} a_1, a_2, ..., a_n \end{bmatrix} \cdot \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix} \cdot \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ \cdot \\ C_n \end{bmatrix} \tag{10.9}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot d_{ij} \cdot c_j \tag{10.10}$$

Reference [2] contains several numerical examples of how to perform system effectiveness calculations using the WSEIAC model. Also, Ref. [3], Chapter VII, discusses the model at length and provides numerical examples.

10.2.3 The Navy Concept of System Effectiveness (Ref. [4])

In the early 1960's, the Navy developed a system effectiveness concept which also combines three basic system characteristics: performance, availability and utilization. It can be expressed as "a measure of the extent to which a system can be expected to complete its assigned mission within an established time frame under stated environmental conditions." It may also be defined mathematically as "the probability that a system can successfully meet an operational demand through a given time period when operated under specified conditions."

Mathematically it has been formulated as follows:

$$E_S = PAU \tag{10.11}$$

where:

E_S = index of system effectiveness

P = index of system performance - a numerical index expressing system capability, assuming a hypothetical 100% availability and utilization of performance capability in actual operation

A = index of the system availability - a numerical index of the extent to which the system is ready and capable of fully performing its assigned mission(s)

U = index of system utilization - a numerical index of the extent to which the performance capability of the system is utilized during the mission

The components of the Navy model are not as readily computed as are those of the ARINC and WSEIAC models. The Navy has stated that the terms PU and A are similar to the WSEIAC terms C and AD (Ref. [5]) and that PAU can be translated into the analytical terms P_C and P_T

where:

P_C **performance capability** - a measure of adequacy of design and system degradation

P_T **detailed time dependency** - a measure of availability with a given utilization

Thus the Navy model is compatible with the WSEIAC model in the following way:

$$f(PAU) = f(P_C , P_T) = f(A,D,C) \tag{10.12}$$

The WSEIAC, Navy and ARINC concepts of system effectiveness are depicted in Figure 10.2-1.

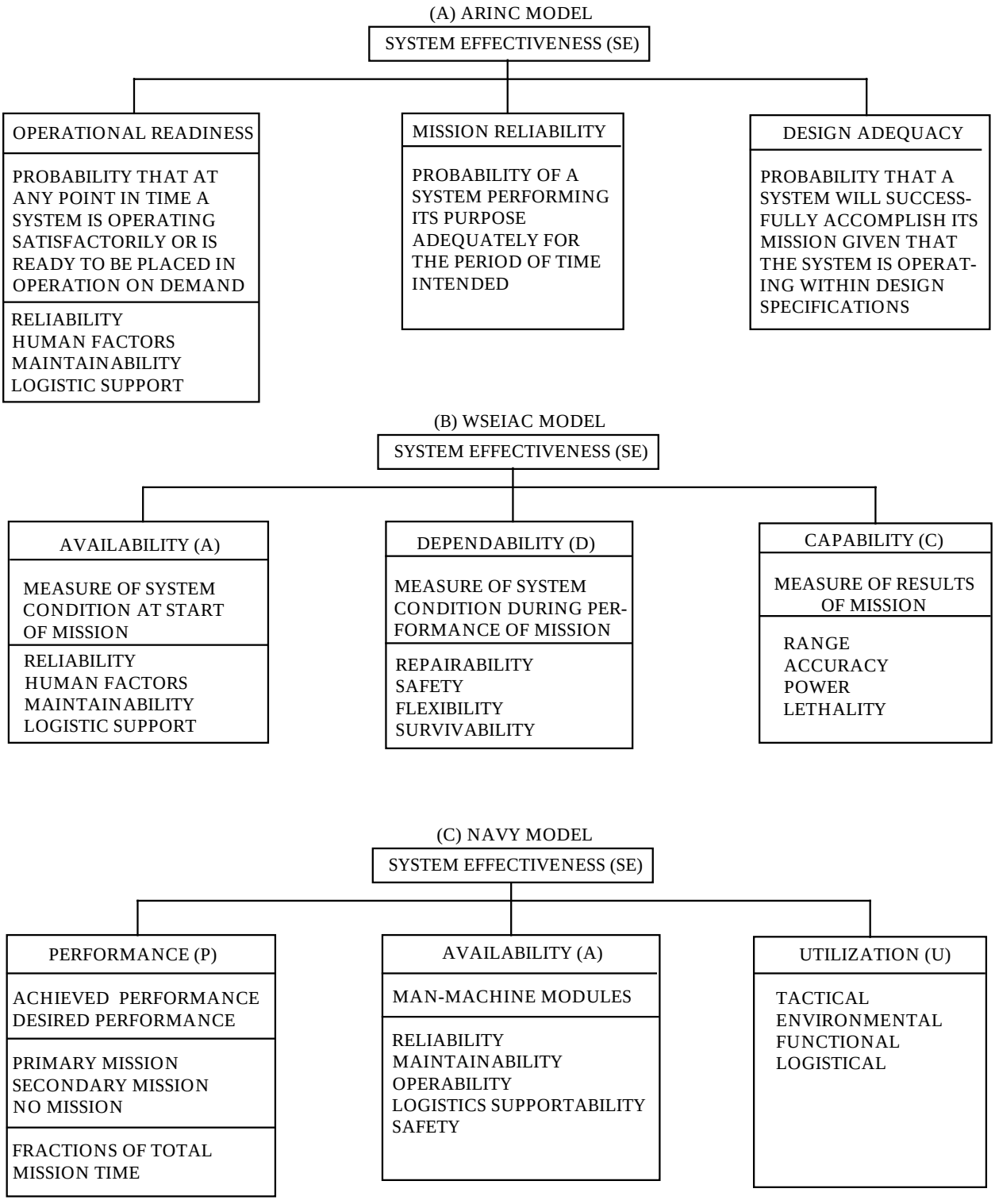


FIGURE 10.2-1: SYSTEM EFFECTIVENESS MODELS

Although these models are relatively simple to describe, their development and application is a rather complex process usually performed by operations research groups and operations analysts utilizing available computerized models (to be discussed later).

10.2.4 An Illustrative Model of a System Effectiveness Calculation

The following simplified example, utilizing the WSEIAC concept, is provided in order to show how R&M parameters are used in system effectiveness calculations.

Problem Statement

The system to be considered consists of a helicopter and its communication equipment. It is to operate in a limited warfare environment where rapid movement of supplies upon request is important. The mission of the system is that upon random call of transporting supplies from a central supply to operational activities within a radius of one-half hour flying time and providing vertical underway replenishment of needed spares. Once the helicopter has reached the target area, proper functioning of the communication equipment enhances the chances of a successful delivery of the supplies in terms of safe delivery, timely delivery, etc. Some major assumptions which are inherent in this example are:

(1) A call for supplies is directed to a single helicopter. If this craft is not in flyable condition (i.e., it is in process of maintenance), the mission will not be started. A flyable craft is defined as one which is in condition to take off and fly with a standard supply load.

(2) The flight time required to reach the target area is one-half hour.

(3) The communication equipment cannot be maintained or repaired in flight.

(4) A loaded helicopter which goes down while enroute to, or which does not reach, the target area, has no delivery value.

Model Determination

For purposes of model formulation, the system condition is divided into three states:

(1) State 1: Helicopter flyable, communication equipment operable

(2) State 2: Helicopter flyable, communication equipment nonoperable

(3) State 3: Helicopter nonflyable

The WSEIAC model for effectiveness is given by the following equation:

$$SE = ADC$$

where A, D and C are defined as follows:

(1) The availability vector is a three-element, row vector, i.e.,

$$A = (a_1 , a_2 , a_3)$$

where a_i is the probability that the helicopter will be in State i at the time of call.

(2) The dependability matrix is a 3x3 square matrix, i.e.,

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}$$

where d_{ij} is the probability that if the helicopter is in State i at the time of call it will complete the mission in State j.

(3) The capability vector is a three-element column vector, i.e.,

$$C = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

where c_i is the probability that if the helicopter is in State i at the time of arrival at the target area the supplies can be successfully delivered. (For multi-capability items, C would be a multi-column matrix.)

Determination of Model Elements

Past records indicate that the average time between maintenance activities (including preventive and failure initiated maintenance) for this type of helicopter is 100 hours and the average duration (including such variables as maintenance difficulty, parts availability, manpower, etc.) of a maintenance activity is ten hours. Comparable data for the communication equipment shows an average time between maintenance activities of 500 hours and an average duration of a maintenance activity of five hours.

From the preceding data the elements of A can be determined.

A_1 = P(helicopter flyable) • P(communication equipment operable)

$$= \left(\frac{100}{100 + 10}\right)\left(\frac{500}{500 + 5}\right) = 0.9$$

A_2 = P(helicopter flyable) • P(communication equipment not operable)

$$= \left(\frac{100}{100 + 10}\right)\left(\frac{5}{500 + 5}\right) = 0.009$$

A_3 = P(helicopter not flyable) $= \left(\frac{10}{100 + 10}\right) = 0.091$

Data from past records indicates that the time between failures of the communication equipment during flight is exponentially distributed with a mean of 500 hours. Also, the probability that a helicopter in flight will not survive the one-half hour flight to its destination is 0.05 (includes probability of being shot down, mechanical failures, etc.). Then the elements of the D matrix may be calculated as follows:

(1) If the system begins in State 1:

d_{11} = P(helicopter will survive flight) • P(communication equipment will remain operable)

$$= (1 - 0.05)\ \exp\left[\left(-\frac{1/2}{500}\right)\right] = 0.94905$$

d_{12} = P(helicopter will survive flight) • P(communication equipment will fail during flight)

$$= (1 - 0.05)\ \left[1 - \exp\left(-\frac{1/2}{500}\right)\right] = 0.00095$$

d_{13} = P(helicopter will not survive the flight) = 0.05000

(2) If the system begins in State 2:

d_{21} = 0 because the communication equipment cannot be repaired in flight

d_{22} = P(helicopter will survive flight) = 0.95000

d_{23} = P(helicopter will not survive the flight) = 0.05000

(3) If the system begins in State 3:

d_{31} = d_{32} = 0 because the mission will not start

d_{33} = 1, i.e., if the helicopter is not flyable, it will remain nonflyable with reference to a particular mission

Experience and technical judgment have determined the probability of successful delivery of supplies to be c_i if the system is in State i at the time of arrival in the target area, where

$c_1 = 0.95 \quad c_2 = 0.80 \quad c_3 = 0$

Determination of Effectiveness

The effectiveness of the subject system becomes

$$E = \begin{bmatrix} 0.900 & 0.009 & 0.091 \end{bmatrix} \begin{bmatrix} 0.94905 & 0.00095 & 0.05 \\ 0 & 0.95 & 0.05 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.95 \\ 0.8 \\ 0 \end{bmatrix} = 0.82$$

which means that the system has a probability of 0.82 of successful delivery of supplies upon random request.

The effectiveness value attained provides a basis for deciding whether improvement is needed. The model also provides the basis for evaluating the effectiveness of alternative systems considered.

SECTION 10: SYSTEMS RELIABILITY ENGINEERING

10.3 System R&M Parameters

In this section we are concerned with those system effectiveness submodels, e.g., availability, dependability, operational readiness, which can be exercised to specify, predict, allocate, optimize, and measure system R&M parameters.

Four types of parameters and examples of specific R&M terms applicable to their specification and measurement, are shown in Table 10.3-1. Each will be discussed in more detail in the following paragraphs.

TABLE 10.3-1: SYSTEM R&M PARAMETERS

OBJECTIVES	EXAMPLE TERMS
• READINESS OR AVAILABILITY	R: Mean Time Between Downing Events M: Mean Time to Restore System
• MISSION SUCCESS	R: Mission Time Between Critical Failures M: Mission Time to Restore Function
• MAINTENANCE MANPOWER COST	R: Mean Time Between Maintenance Actions M: Direct Man-hours per Maintenance Action
• LOGISTIC SUPPORT COST	R: Mean Time Between Removals M: Total Parts Cost per Removal

Operational Readiness R&M Parameters - These parameters will define the R&M contribution to the readiness measurement of the system or unit. R&M by itself does not define readiness; there are many other factors relating to personnel, training, supplies, etc., that are necessarily included in any real measure of readiness. The context of readiness includes many factors beyond the realm of equipment capability and equipment R&M achievements. R&M parameters of this type concern themselves with the likelihood of failures occurring that would make a ready system no longer ready and with the effort required to restore the system to the ready condition. Examples of this type of parameter are “mean time between downing events” for reliability and “mean time to restore system” for maintainability.

Mission Success R&M Parameters - These parameters are similar to the classical reliability discussion that is found in most reliability text books. They relate to the likelihood of failures occurring during a mission that would cause a failure of that mission and the efforts that are directed at correcting these problems during the mission itself. Examples would be “mission time between critical failures (MTBCF)” for reliability and “mission time to restore function” for maintainability.

Maintenance Manpower Cost R&M Parameters - Some portion of a system's maintenance manpower requirement is driven by the system's R&M achievement. This category of system R&M parameters concerns itself with how frequently maintenance manpower is required and,

once it is required, how many man-hours are needed. Examples of this type of parameter are "mean time between maintenance actions" for reliability and "direct man-hours to repair" for maintainability. Note that the maintainability example does not address the clock hours to complete the repair. Time to restore the system, i.e., the system downtime, is not as significant to the people concerned with manpower needs as the total man-hours required.

Logistic Support Cost R&M Parameters - In many systems, this type of R&M parameter might be properly titled as "material cost" parameters. These parameters address the aspect of R&M achievement that requires the consumption of material. Material demands also relate to the readiness or availability of the system. Examples are "mean time between removals" for reliability and "total parts cost per removal" for maintainability.

Let us examine some of the techniques for using reliability data, reduced to parameters such as those just discussed, for making reliability predictions.

10.3.1 Parameter Translation Models

Frequently it is necessary to convert various reliability parameters from one set of environmental conditions to a different set of environmental conditions. Extensive reliability data may have been generated or gathered in a given environment while the equipment may be subsequently slated for use in an entirely different environment. In other cases, the customers may define a reliability parameter differently than the manufacturer does, or he may use an entirely different figure-of-merit as the basis for acceptance. The intent of this section is to address these areas of concern.from:kekaoxing.com

10.3.1.1 Reliability Adjustment Factors

"What if" questions are often asked regarding reliability figures of merit for different operating conditions. For example, what reliability could be expected from a product in a ground fixed environment that is currently experiencing a 700 hour MTBF in an airborne environment. Tables have been derived to make estimates of the effects of quality levels, environments and temperatures enabling rapid conversions between environments. The database upon which these tables are based was a grouping of approximately 18,000 parts from a number of equipment reliability predictions performed on various military contracts. Ratios were developed using this database and the MIL-HDBK-217F algorithms. The relative percentages of each part type in the database are shown in Figure 10.3-1.

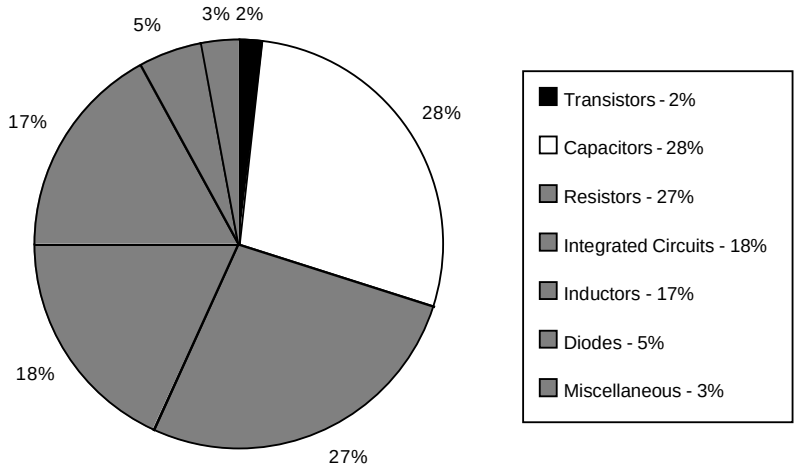


FIGURE 10.3-1: PART DATABASE DISTRIBUTION

(Source: Reliability Toolkit: Commercial Practices Edition, Rome Laboratory and Reliability Analysis Center, Rome, NY 1995).

The following tables, 10.3-2 through 10.3-4, provide a means of converting a known reliability value, expressed as an MTBF, from one set of conditions to another.

TABLE 10.3-2: PART QUALITY FACTORS (MULTIPLY SERIES MTBF BY)

		To Quality Class			
	Part Quality	**Space**	**Military**	**Ruggedized**	**Commercial**
From Quality Class	**Space**	X	0.8	0.5	0.2
	Full Military	1.3	X	0.6	0.3
	Ruggedized	2.0	1.7	X	0.4
	Commercial	5.0	3.3	2.5	X

Space - Extra Testing Beyond Full Military
Military - Standardized 100% Chip Testing
Ruggedized - Selected 100% Chip Testing
Commercial - Vendor Discretion Testing

TABLE 10.3-3: ENVIRONMENTAL CONVERSION FACTORS (MULTIPLY SERIES MTBF BY)

From Environment	To Environment: G_B	G_F	G_M	N_S	N_U	A_{IC}	A_{IF}	A_{UC}	A_{UF}	A_{RW}	S_F
	G_B	X	0.5	0.2	0.3	0.1	0.3	0.2	0.1	0.1	0.1	1.2
	G_F	1.9	X	0.4	0.6	0.3	0.6	0.4	0.2	0.1	0.2	2.2
	G_M	4.6	2.5	X	1.4	0.7	1.4	0.9	0.6	0.3	0.5	5.4
	N_S	3.3	1.8	0.7	X	0.5	1.0	0.7	0.4	0.2	0.3	3.8
	N_U	7.2	3.9	1.6	2.2	X	2.2	1.4	0.9	0.5	0.7	8.3
	A_{IC}	3.3	1.8	0.7	1.0	0.5	X	0.7	0.4	0.2	0.3	3.9
	A_{IF}	5.0	2.7	1.1	1.5	0.7	1.5	X	0.6	0.4	0.5	5.8
	A_{UC}	8.2	4.4	1.8	2.5	1.2	2.5	1.6	X	0.6	0.8	9.5
	A_{UF}	14.1	7.6	3.1	4.4	2.0	4.2	2.8	1.7	X	1.4	16.4
	A_{RW}	10.2	5.5	2.2	3.2	1.4	3.1	2.1	1.3	0.7	X	11.9
	S_F	0.9	0.5	0.2	0.3	0.1	0.3	0.2	0.1	0.1	0.1	X

Environmental Factors as Defined in MIL-HDBK-217

G_B - Ground Benign; G_F - Ground Fixed; G_M - Ground Mobile; N_S - Naval Sheltered; N_U - Naval Unsheltered; A_{IC} - Airborne Inhabited Cargo; A_{IF} - Airborne Inhabited Fighter; A_{UC} - Airborne Uninhabited Cargo; A_{UF} - Airborne Uninhabited Fighter; A_{RW} - Airborne Rotary Winged; S_F - Space Flight

CAUTION: Do not apply to MTBCF.

TABLE 10.3-4: TEMPERATURE CONVERSION FACTORS
(MULTIPLY SERIES MTBF BY)

		To Temperature (°C)						
	Temp.	**10**	**20**	**30**	**40**	**50**	**60**	**70**
	10	X	0.9	0.8	0.8	0.7	0.5	0.4
	20	1.1	X	0.9	0.8	0.7	0.6	0.5
	30	1.2	1.1	X	0.9	0.8	0.6	0.5
From Temperature (°C)	**40**	1.3	1.2	1.1	X	0.9	0.7	0.6
	50	1.5	1.4	1.2	1.1	X	0.8	0.7
	60	1.9	1.7	1.6	1.5	1.2	X	0.8
	70	2.4	2.2	1.9	1.8	1.5	1.2	X

10.3.1.2 Reliability Prediction of Dormant Products

In the past, analysis techniques for determining reliability estimates for dormant or storage conditions relied on simple rules of thumb such as: “the failure rate will be reduced by a ten to one factor”, or “the expected failure rate is zero.” A more realistic estimate, based on part count failure results, can be calculated by applying the conversion factors shown for the example in Table 10.3-5. The factors convert operating failure rates by part type to dormant conditions for seven scenarios.

These conversion factors were determined using data from various military contracts and algorithms from both MIL-HDBK-217F and RADC-TR-85-91, “Impact of Nonoperating Periods on Equipment Reliability” (Ref. [34]). Average values for operating and dormant failure rates were developed for each scenario. For example, to convert the reliability of an operating airborne receiver to a ground nonoperating condition, determine the number of components by type, then multiply each by the respective operating failure rate obtained from handbook data, field data, or vendor estimates. The total operating failure rate for each type is then converted using the conversion factors of Table 10.3-5. The dormant estimate of reliability for the example receiver is determined by summing the part results.

TABLE 10.3-5: AIRCRAFT RECEIVER CONVERSION:
AIRBORNE OPERATING TO GROUND DORMANT FAILURE RATE (EXAMPLE)

Device	Qty	λ_O	λ_{TO}	Conversion Factor	λ_D
Integrated Circuit	25	0.06	1.50	.04	.060
Diode	50	0.001	0.05	.01	.001
Transistor	25	0.002	0.05	.02	.001
Resistor	100	0.002	0.20	.03	.006
Capacitor	100	0.008	0.80	.03	.024
Switch	25	0.02	0.50	.10	.050
Relay	10	0.40	4.00	.04	.160
Transformer	2	0.05	0.10	.20	.020
Connector	3	1.00	3.00	.003	.009
Printed Circuit Board	1	0.70	0.70	.01	.007
Totals	---	---	10.9	---	0.338

λ_O = Part (Operating) Failure Rate (Failures per Million Hours)
λ_{TO} = Total Part (Operating) Failure Rate (Failures per Million Hours)
λ_D = Total Part Dormant Failure Rate (Failures per Million Hours)

Mean-Time-Between-Failure (Operating) = 92,000 hours
Mean-Time-Between-Failure (Dormant) = 2,960,000 hours

10.3.2 Operational Parameter Translation

Field operation typically introduces factors which are beyond the control of designers (e.g. maintenance policy). Thus, “design” reliability may not be the same as “operational” reliability. For this reason, it is often necessary to convert, or translate, from “design” to “operational” terms and vice versa. This translation technique is based on RADC-TR-89-299, “Reliability and Maintainability Operational Parameter Translation II” (Ref. [35]) which developed models for the two most common environments, ground and airborne. While these models are based on military use, similar differences can be expected for commercial products. The translation models are summarized in Table 10.3-6.

TABLE 10.3-6: RELIABILITY TRANSLATION MODELS

	R_F Selection						
	Communication	Navigation	Computer	Counter Measure	Radar	All Other	Dependent Var. Lower Bound (% of Ind. Var.)*
1. Airborne Fighter Models							
1A. $MTBF_F = \theta_P^{.64} R_F \left(\frac{C}{D}\right)^{-.46}$	2.1	6.5	5.9	4.7	3.6	4.3	48
1B. $MTBM_F = \theta_P^{.64} R_F \left(\frac{C}{D}\right)^{-.57}$	1.1	2.7	1.9	2.8	1.7	2.0	24
1C. $MTBR_F = \theta_P^{.62} R_F \left(\frac{C}{D}\right)^{-.77}$	1.8	4.4	3.0	5.9	2.5	3.2	34
1D. $MTBF_F = \theta_D^{.76} R_F \left(\frac{C}{D}\right)^{-.34}$	2.1	5.0	5.3	3.7	5.1	2.2	79
1E. $MTBM_F = \theta_D^{.75} R_F \left(\frac{C}{D}\right)^{-.44}$	1.4	2.2	1.8	2.4	2.8	.90	36
1F. $MTBR_F = \theta_D^{.77} R_F \left(\frac{C}{D}\right)^{-.65}$	1.6	4.0	2.2	3.4	3.0	.83	49
2. Airborne Transport Models	**R_F, Uninhabited Equipment**			**R_F, Inhabited Equipment**			
2A. $MTBF_F = \theta_P^{.73} R_F \left(\frac{C}{D}\right)^{-.46}$	2.7			2.5			50
2B. $MTBM_F = \theta_P^{.69} R_F \left(\frac{C}{D}\right)^{-.57}$	1.6			1.4			26
2C. $MTBR_F = \theta_P^{.66} R_F \left(\frac{C}{D}\right)^{-.77}$	2.1			2.3			35
2D. $MTBF_F = \theta_D^{1.0} R_F \left(\frac{C}{D}\right)^{-.34}$	.58			.39			91
2E. $MTBM_F = \theta_D^{1.1} R_F \left(\frac{C}{D}\right)^{-.44}$	.13			.09			44
2F. $MTBR_F = \theta_D^{.88} R_F \left(\frac{C}{D}\right)^{-.65}$	.78			.60			72
3. Ground System Models	**R_F, Fixed Equipment**			**R_F, Mobile Equipment**			
3A. $MTBF_F = \theta_P^{.60} R_F$	27			4.8			90
3B. $MTBM_F = \theta_P^{.67} R_F$	11			1.8			49
3C. $MTBR_F = \theta_P^{.50} R_F$	91			18			80

*The field numeric (i.e., MTBFF, MTBMF or MTBRF) is always taken to be the lesser of (1) the calculated value from Column 1 or, (2) the percentage shown of the independent variable (i.e., θ_P or θ_D).

10.3.2.1 Parameter Definitions

- Mean-time-between-failure-field ($MTBF_F$) includes inherent maintenance events which are caused by design and manufacturing defects.

$$MTBF_F = \frac{\text{Total Operating Hours or Flight Hours}}{\text{Inherent Maintenance Events}}$$

- Mean-time-between-maintenance-field ($MTBM_F$) consists of inherent, induced and no defect found maintenance actions.

$$MTBM_F = \frac{\text{Total Operating Hours or Flight Hours}}{\text{Total Maintenance Events}}$$

- Mean-time-between-removals-field ($MTBR_F$) includes all removals of the equipment from the system.

$$MTBR_F = \frac{\text{Total Operating Hours or Flight Hours}}{\text{Total Equipment Removals}}$$

- θ_P = the predicted MTBF (i.e., estimated by failure rates of the part population)
- θ_D = the demonstrated MTBF (i.e., controlled testing)
- R_F = the equipment type or application constant
- C = the power on-off cycles per mission or operating event
- D = the mission duration or operating event

10.3.2.2 Equipment Operating Hour to Flight Hour Conversion

For airborne categories - $MTBF_F$ represents the mean-time-between-failure in equipment operating hours. To obtain $MTBF_F$ in terms of flight hours (for both fighter and transport models), divide $MTBF_F$ by 1.2 for all categories except countermeasures. Divide by .8 for countermeasures equipment.

Example 1:

Estimate the MTBM of a fighter radar given a mission length of 1.5 hours, two radar shutdowns per mission and a predicted radar MTBF of 420 hours. Using Model 1B in Table 10.3-6,

$$MTBM_F = \theta_P^{.64}\ R_F \left(\frac{C}{D}\right) - .57 = (420\text{ hr.})^{.64}\ 1.7 \left(\frac{2\text{ cyc.}}{1.5\text{ hr.}}\right) - .57$$

$MTBM_F$ = 69 equipment operating hours between maintenance.

Since this is below the dependent variable lower bound of (.24)(420) = 101 hours, the calculated $MTBM_F$ is correct. Since this equipment is often turned on for pre- and post-flight checkout, the number of flight hours between maintenance is somewhat less than the actual equipment operating hours. The number of flight hours between maintenance is approximately 69/1.2 = 58 hours.

Example 2:

Estimate the MTBF of a commercial airline navigation unit used on an 8 hour flight and shut down after the flight. The predicted MTBF for the navigation unit is 2,000 hours. Using model 2A for inhabited environment,

$$MTBF_F = \theta_P^{.73} \; R_F \left(\frac{C}{D}\right)^{.46}$$

$$= (2{,}000)^{.73} \; 2.5 \left(\frac{1 \text{ cycle}}{8 \text{ hours}}\right)^{.46}$$

$MTBF_F$ = 1,672 hours between failure

The number of flight hours between failure is estimated to be 1,672/1.2 = 1,393 hours. However, in accordance with the footnote of Table 10.3-6, we calculate a value of (.50)(2000) = 1000 hours using the dependent variable bound. Since this is less than the previous calculation, this is the value to be used.

10.3.3 Availability, Operational Readiness, Mission Reliability, and Dependability - Similarities and Differences

As can be seen from their definitions in Table 10.3-7, availability and operational readiness refer to the capability of a system to perform its intended function when called upon to do so. This emphasis restricts attention to probability “at a point in time” rather than “over an interval of time.” Thus, they are point concepts rather than interval concepts. To differentiate between the two: availability is defined in terms of operating time and downtime, where downtime includes active repair time, administrative time, and logistic time; whereas, operational readiness includes all of the availability times plus both free time and storage time, i.e., all calendar time.

TABLE 10.3-7: DEFINITIONS OF KEY R&M SYSTEM PARAMETERS

AVAILABILITY: A measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at an unknown (random) time. (Item state at start of a mission includes the combined effects of the readiness-related system R&M parameters but excludes mission time.)

OPERATIONAL READINESS: The ability of a military unit to respond to its operation plan(s) upon receipt of an operations order. (A function of assigned strength, item availability, status or supply, training, etc.)

MISSION RELIABILITY: The ability of an item to perform its required functions for the duration of a specified "mission profile."

DEPENDABILITY: A measure of the degree to which an item is operable and capable of performing its required function at any (random) time during a specified mission profile, given item availability at the start of the mission. (Item state during a mission includes the combined effects of the mission-related system R&M parameters but excludes non-mission time.) (This definition is different than the definition of dependability as it appears in IEC documents.)

MEAN-TIME-BETWEEN-DOWNING-EVENTS (MTBDE): A measure of the system reliability parameter related to availability and readiness. The total number of system life units divided by the total number of events in which the system becomes unavailable to initiate its mission(s) during a stated period of time.

MEAN-TIME-TO-RESTORE-SYSTEM (MTTRS): A measure of the system maintainability parameters related to availability and readiness: the total corrective maintenance time associated with downing events divided by the total number of downing events during a stated period of time. (Excludes time for off-system maintenance and repair of detached components.)

MISSION-TIME-BETWEEN-CRITICAL-FAILURES (MTBCF): A measure of mission reliability: the total amount of mission time divided by the total number of critical failures during a stated series of missions.

MISSION-TIME-TO-RESTORE-FUNCTIONS (MTTRF): A measure of mission maintainability: the total corrective critical failure maintenance time divided by the total number of critical failures during the course of a specified mission profile.

MEAN-TIME-BETWEEN-MAINTENANCE-ACTIONS (MTBMA): A measure of the system reliability parameter related to demand for maintenance manpower: the total number of system life units divided by the total number of maintenance actions (preventive and corrective) during a stated period of time.

DIRECT-MAINTENANCE-MAN-HOURS-PER-MAINTENANCE-ACTION (DMMH/MA): A measure of the maintainability parameter related to item demand for maintenance manpower: the sum of direct maintenance man-hours divided by the total number of maintenance actions (preventive and corrective) during a stated period of time.

Also note that the concepts of availability and operational readiness do not include mission time.

Dependability, although it is a point concept like availability and operational readiness, differs from those concepts in that it is concerned with the degree (or probability) that an item is operable at some point (time) during the mission profile, given its (point) availability at the start of the mission.

Mission reliability, on the other hand, is concerned with the ability of a system to continue to perform without failure for the duration of a specified mission time; in other words, the probability of successful operation over some interval of time rather than at a specific point in time. Thus, mission reliability is an interval concept rather than a point concept. It should be pointed out that mission reliability is also conditional upon the system being operable at the beginning of the mission or its (point) availability.

Further note that dependability and mission reliability do not include non-mission time.

Hopefully, the mathematical models and examples which follow will help to further clarify these concepts.

10.4 System, R&M Modeling Techniques

It was previously pointed out in Section 5 that mathematical models represent an efficient, shorthand method of describing an event and the more significant factors which may cause or affect the occurrence of the event. Such models are useful to engineers and designers since they provide the theoretical foundation for the development of an engineering discipline and a set of engineering design principles which can be applied to cause or prevent the occurrence of an event.

At the system level, models such as system effectiveness models (and their R&M parameter submodels) serve several purposes:

(1) To evaluate the effectiveness of a system of a specific proposed design in accomplishing various operations (missions) for which it is designed and to calculate the effectiveness of other competing designs, so that the decision maker can select that design which is most likely to meet specified requirements,

(2) To perform trade-offs among system characteristics, performance, reliability, maintainability, etc., in order to achieve the most desirable balance among those which result in highest effectiveness,

(3) To perform parametric sensitivity analyses in which the numerical value of each parameter is varied in turn and to determine its effect on the numerical outputs of the model. Parameters that have little or no effect can be treated as constants and the model simplified accordingly. Parameters to which the model outputs show large sensitivity are then examined in detail, since small improvements in the highly sensitive

parameters may result in substantial improvements in system effectiveness at very acceptable cost,

(4) To "flag" problem areas in the design which seriously limit the ability of the design to achieve the desired level of system R&M or system effectiveness.

The evaluation of system effectiveness and its R&M parameters is an iterative process that continues through all life cycle phases of a system. In each of these phases, system effectiveness is continually being "measured" by exercising the system effectiveness models. In the early design stage, system effectiveness and R&M predictions are made for various possible system configurations. When experimental hardware is initially tested, first real life information is obtained about performance, reliability, and maintainability characteristics, and this information is fed into the models to update the original prediction and to further exercise the models in an attempt to improve the design. This continues when advanced development hardware is tested to gain assurance that the improvements in the system design are effective or to learn what other improvements can still be made before the system is fully developed, type classified, and deployed for operational use. Once in operation, field data starts to flow in and the models are then used to evaluate the operational effectiveness of the system as affected by the field environment, including the actual logistic support and maintenance practices provided in the field. The models again serve to disclose or "flag" problem areas needing improvement.

One may summarize the need for system R&M models as follows:

> They provide insight, make an empirical approach to system design and synthesis economically feasible, and are a practical method for circumventing a variety of external constraints. Furthermore, the models aid in establishing requirements, provide an assessment of the odds for successful mission completion, isolate problems to definite areas, and rank problems to their relative seriousness of impact on the mission. They also provide a rational basis for evaluation and choice of proposed system configurations and for proposed solutions to discovered problems.

Thus, system R&M models are an essential tool for the quantitative evaluation of system effectiveness and for designing effective weapon systems. Figure 10.4-1 identifies eight principal steps involved in system effectiveness evaluation. Step 1 is mission definition, Step 2 is system description, Step 3 is selection of figure of merit, and Step 4 is the identification of accountable factors that impose boundary conditions and constraints on the analysis to be conducted. After completing these four Steps, it becomes possible to proceed with Step 5, the construction of the mathematical models. To obtain numerical answers from the models, numerical values of all parameters included in the models must be established or estimated (Step 7). To do this, good and reliable data must first be acquired from data sources, tests, etc. (Step 6). In the final Step 8, the models are exercised by feeding in the numerical parametric values to obtain system effectiveness estimates and to perform optimizations. Ref. [7] illustrates in more detail the whole process of system effectiveness evaluations, beginning with the military

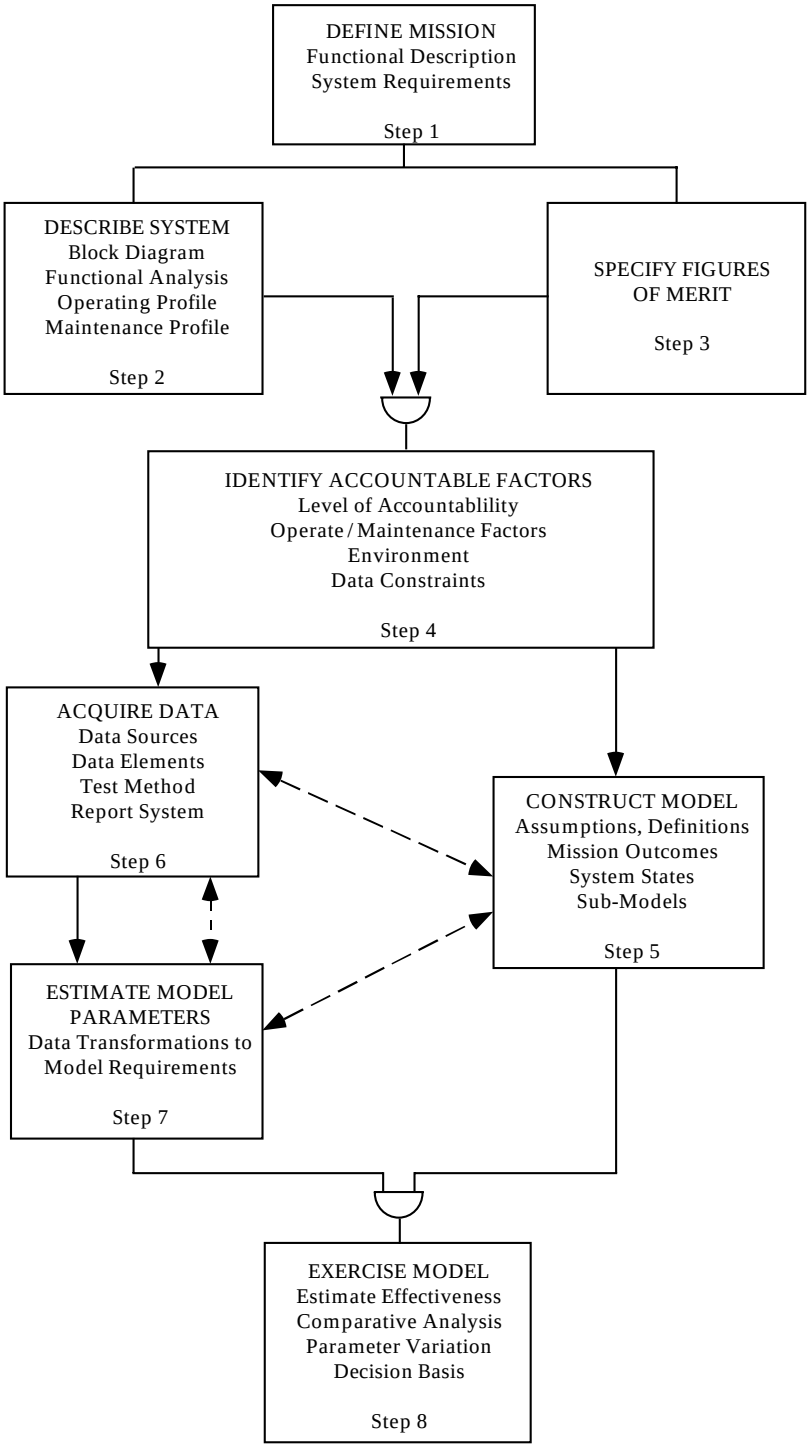


FIGURE 10.4-1: PRINCIPAL STEPS REQUIRED FOR EVALUATION OF SYSTEM EFFECTIVENESS

operational requirements and progressing through the exercising of the system effectiveness model(s) to the decision-making stage.

In terms of system R&M parameter models, reliability and maintainability define system availability and/or operational readiness. Reliability determines the state probabilities of the system during the mission, i.e., the system dependability. If repairs can be performed during the mission, maintainability also becomes a factor in dependability evaluations; this case is often referred to as "reliability with repair." Then, there is the impact of logistic support on the downtime and turnaround time of the system, since shortcomings in the logistic support may cause delays over and above the maintenance time as determined by the system maintainability design. Finally, there are the performance characteristics of the system that are affected by the state in which the system may be at any point in time during a mission, i.e., by the system dependability.

Submodels of availability, operational readiness, downtime distributions, dependability, etc., are required to obtain the numerical answers that may be fed into an overall system effectiveness model, if such can be constructed. Some of these submodeling techniques will now be discussed.

10.4.1 Availability Models

The concept of availability was originally developed for repairable systems that are required to operate continuously, i.e., round-the-clock, and are at any random point in time either operating or "down" because of failure and are being worked upon so as to restore their operation in minimum time. In this original concept, a system is considered to be in only two possible states -- operating or in repair -- and availability is defined as the probability that a system is operating satisfactorily at any random point in time, t, when subject to a sequence of "up" and "down" cycles which constitute an alternating renewal process.

Availability theory was treated quite extensively in Section 5; this section will concentrate on final results and illustrative examples of the various models.

10.4.1.1 Model A - Single Unit System (Point Availability)

Consider first a single unit system or a strictly serial system that has a reliability, R(t); its availability, A(t), that it will be in an "up" state (i.e., will be operating) at time, t, when it started in an "up" condition at t = 0 is given by:

$$A(t) = \frac{\mu}{\lambda + \mu} + \left\{ \frac{\lambda}{\lambda + \mu} \exp\left[-(\lambda + \mu)t \right] \right\} \quad (10.13)$$

where:

λ is the failure rate and μ is the repair rate

If it started in a "down" state at t = 0

$$A(t) = \frac{\mu}{\lambda + \mu} - \left\{ \frac{\lambda}{\lambda + \mu} \exp \left[-(\lambda + \mu)t \right] \right\} \tag{10.14}$$

This assumes that the probability density functions for failures and repairs are exponentially distributed and given by, respectively:

$$f(t) = \lambda e^{-\lambda t} \tag{10.15}$$

$$g(t) = \mu e^{-\lambda t} \tag{10.16}$$

We may write Equation 10.13 also in terms of the reciprocal values of the failure and repair rates, i.e., in terms of the MTBF and the MTTR, remembering, however, that both time-to-failure and time-to-repair must be exponentially distributed for the equation to hold.

$$A(t) = \frac{MTBF}{MTBF + MTTR} + \left\{ \frac{MTTR}{MTBF + MTTR} \bullet \exp \left[-\left(\frac{1}{MTBF} + \frac{1}{MTTR} \right) t \right] \right\} \tag{10.17}$$

When we study this equation we see that as t increases the second term on the right diminishes and that availability in the limit becomes a constant, i.e.,

$$\lim_{t \to \infty} A(t) = A_s = \frac{MTBF}{MTBF + MTTR} \tag{10.18}$$

We call this the steady-state availability or inherent uptime ratio of a serial system. It is equivalent to the intrinsic availability, A_i, discussed in Section 5.

Figure 10.4-2 shows plots of A(t), instantaneous availability, and A_i or A_S (steady state availability) for a single system having a failure rate, (λ), of 0.01 failures/hour and a repair rate (μ), of 1 repair/hour.

Note that the transient term decays rather rapidly; it was shown in Section 5 that the transient term becomes negligible for

$$t \geq \frac{4}{\lambda + \mu} \tag{10.19}$$

An important point to be made is that Eq. (10.18) holds regardless of the probability distribution of time-to-failure and time-to-repair.

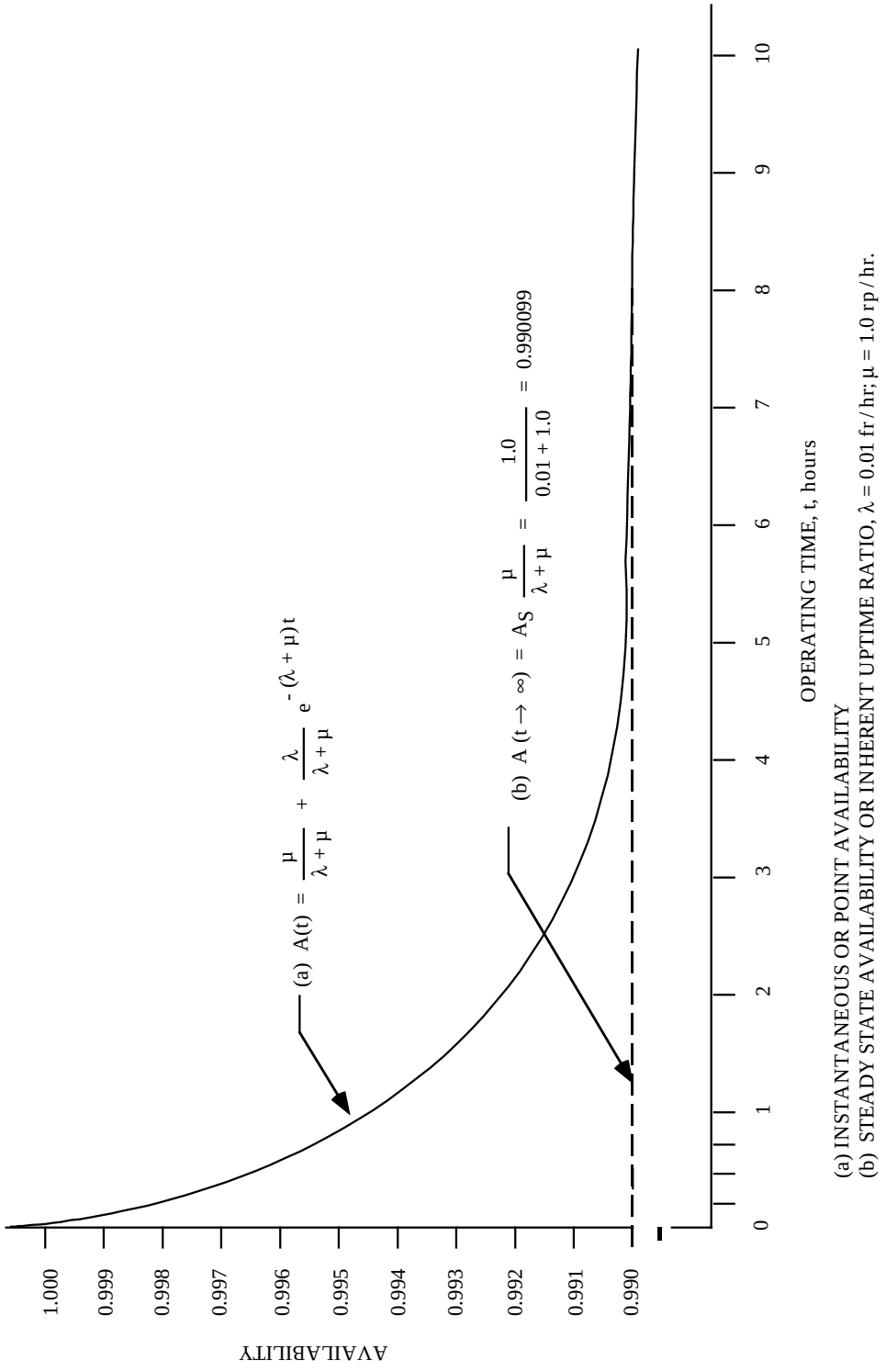


FIGURE 10.4-2: THE AVAILABILITY OF A SINGLE UNIT

Looking again at Eq. (10.18), we may divide the numerator and denominator by the MTBF and write the steady state availability as follows:

$$A = 1/(1 + \alpha) \quad (10.20)$$

where:

α = MTTR/MTBF, the maintenance time ratio (MTR), or alternatively,

α = λ/μ which the reader may recognize from queuing theory as the "utilization" factor. Thus, the availability, A, does not depend upon the actual values of MTBF or MTTR or their reciprocals but only on their ratio.

Since there is a whole range of MTBF ($1/\lambda$) and MTTR ($1/\mu$) values which can satisfy a given availability requirement, the system designer has the option of trading off MTBF and MTTR to achieve the required system availability within technological and cost constraints. This will be discussed later.

Another observation to be made from Eq. (10.20) is that if α, which is equal to MTTR/MTBF, or λ/μ, is less than 0.10, then A_i can be approximated by 1 - MTTR/MTBF, or 1 - λ/μ.

Thus far we have discussed inherent or intrinsic availability which is the fundamental parameter used in equipment/system design. However, it does not include preventive maintenance time, logistic delay time, and administrative time. In order to take these factors into account, we need several additional definitions of availability.

For example, achieved availability, A_a, includes preventive maintenance and is given by the formula:

$$A_a = \frac{MTBM}{MTBM + \overline{M}} \quad (10.21)$$

where $\overline{M}$ is the mean active corrective and preventive maintenance time and MTBM is the mean interval between corrective and preventive maintenance actions equal to the reciprocal of the frequency at which these actions occur, which is the sum of the frequency or rate (λ) at which corrective maintenance actions occur and the frequency or rate (f) at which preventive maintenance actions occur.

Therefore,

$$MTBM = 1/(\lambda + f)$$

Operational availability, A_o, includes, in addition to A_a, logistic time, waiting time, and administrative time, so that the total mean downtime MDT becomes:

$$MDT = \overline{M} + \text{Mean Logistic Time} + \text{Mean Administrative Time}$$

and adds to the uptime the ready time, RT, i.e.,

$$A_o = \frac{MTBM + RT}{MTBM + RT + MDT} \tag{10.22}$$

It is important to realize that RT is the system average ready time (available but not operating) in a complete operational cycle, the cycle being MTBM + MDT + RT.

Example 3: Illustration of Availability Calculations

The following example is provided to clarify the concepts in the subsection. A ground radar system was found to have the following R&M parameters. Determine A_i, A_a, and A_o:

MTBF = 100 hours

MTTR = 0.5 hour

Mean active preventive maintenance time = 0.25 hours

Mean logistic time = 0.3 hour

Mean administrative time = 0.4 hours

MTBM = 75 hours for either corrective or preventive maintenance actions

Mean ready time = 20 hours

Intrinsic or Inherent Availability = A_i

$$A_i = \frac{MTBF}{MTBF + MTTR} = \frac{100}{100 + 0.5} = 0.995$$

Achieved Availability = A_a

$$A_a = \frac{MTBM}{MTBM + \overline{M}} = \frac{75}{75 + 0.5 + 0.25} = 0.99$$

Operational Availability = A_o

$$A_o = \frac{MTBM + RT}{MTBM + RT + MDT} = \frac{75 + 20}{75 + 20 + 0.5 + 0.25 + 0.3 + 0.4} = \frac{95}{96.45} = 0.985$$

10.4.1.2 Model B - Average or Interval Availability

What we discussed in the previous section is the concept of point availability which is the probability that the system is "up" and operating at any point in time. Often, however, one may be interested in knowing what percent or fraction of a time interval (a,b) a system can be expected to operate. For example, we may want to determine the availability for some mission time. This is called the interval or average availability, A_{AV} , of a system and is given by the time average of the availability function A(t) averaged over the interval (a,b):

$$A_{AV}(a,b) = \left[1/(b-a) \right] \int_b^a A(t)\, dt \qquad (10.23)$$

For instance, if we want to know the fraction of time a system such as shown in Figure 10.4-2 will be operating counting from t = 0 to any time, T, we substitute A(t) of Eq. (10.13) into Eq. (10.23) and perform the integration. The result is:

$$A_{AV}(T) = \frac{1}{T}\left[\int_0^T \frac{m}{1+m}\, dt + \int_0^T \frac{l}{1+m} \exp\left[-(1+m)t\right] dt \right] \qquad (10.24)$$

$$= \frac{\mu}{\lambda + \mu} + \frac{\lambda}{T(\lambda + \mu)^2} \left\{ 1 - \exp\left[-(\lambda + \mu)T \right] \right\}$$

Figure 10.4-3 shows the relationship of A(t) to $A_{AV}(t)$ for the exponential case. Note that in the limit in the steady state we again get the availability A of Eq. (10.18), i.e.,

$$\lim_{t \to \infty} A_{AV}(t) = \mu / (\lambda + \mu) = \frac{MTBF}{MTBF + MTTR} \qquad (10.25)$$

But in the transient state of the process, as shown in the figure for an interval (0, T), before equilibrium is reached $A_{AV}(t)$ is in the exponential case larger than A(t) for an interval (0, t). This is not true for all distributions, since A(t) and $A_{AV}(t)$ may be subject to very large fluctuations in the transient state.

From Eq. (10.24) we may also get the average or expected "on" time in an interval (0, t) by multiplying $A_{AV}(t)$ and t, the length of the time interval of interest. Ref. [8], pp. 74-83,

contains an excellent mathematical treatment of the pointwise and interval availability and related concepts.

Unavailability (U) is simply one minus availability (1-A).

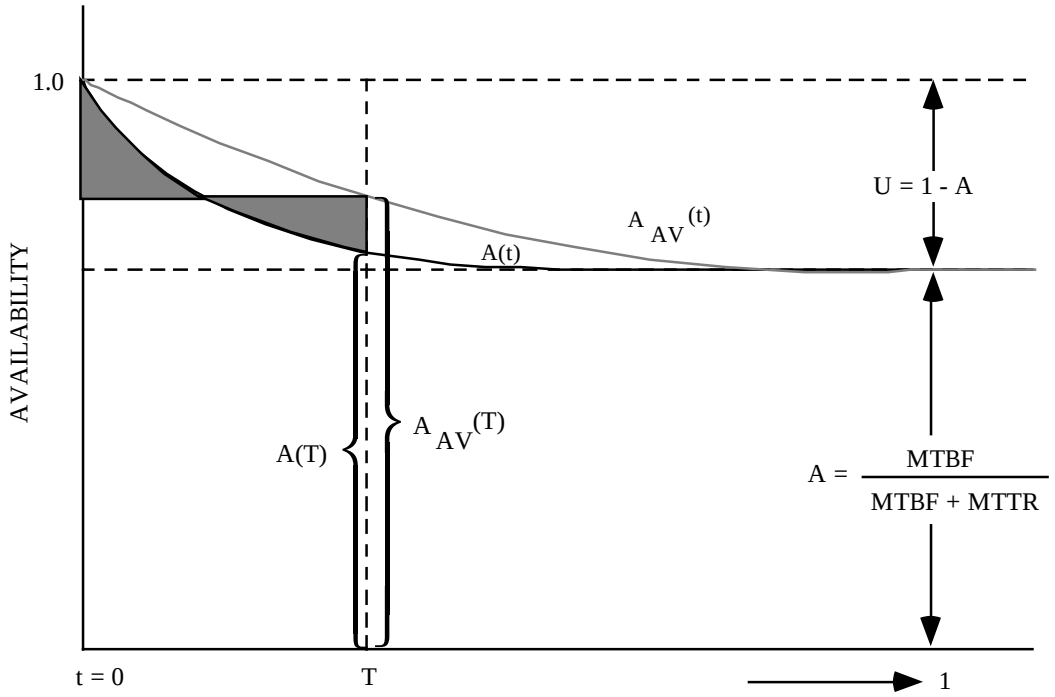


FIGURE 10.4-3: AVERAGE AND POINTWISE AVAILABILITY

Example 4: Average Availability Calculation

Using our ground radar example from the previous subsection, calculate A_{AV} for a mission time of 1 hour.

MTBF = 100 hrs. = $1/\lambda$

MTTR = 0.5 hr. = $1/\mu$

T = 1 hr.

$$A_{AV}(T) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{T(\lambda+\mu)^2}\{1 - \exp[-(\lambda+\mu)T]\}$$

$$= \frac{2}{2.01} + \frac{0.01}{1(2.01)^2}\{1 - \exp[-(2.01)(1)]\}$$

$$= \quad 0.995 + 0.0025\ (1 - 0.134)$$

$$= \quad 0.9972$$

and its expected "on" time for a 1-hr. mission would be (0.9972)(60) = 59.8 minutes.

10.4.1.3 Model C - Series System with Repairable/Replaceable Units

When a series system consists of N units (with independent unit availabilities) separately repairable or replaceable whenever the system fails because of any one unit failing, the steady state availability is given by:

$$A = \prod_{i=1}^{N} A_i \tag{10.26}$$

$$= \prod_{i=1}^{N} \left(\frac{1}{1 + \frac{MTTR_i}{MTBF_i}} \right) \tag{10.27}$$

$$= \prod_{i=1}^{N} \left(\frac{1}{1 + \lambda_1 / \mu_i} \right) \tag{10.28}$$

$$= \prod_{i=1}^{N} \left(\frac{1}{1 + \alpha_i} \right) \tag{10.29}$$

where:

$$\alpha_i = \frac{MTTR_i}{MTBF_i} = \frac{\lambda_i}{\mu_i}$$

Furthermore, if each $\frac{MTTR_i}{MTBF_i}$ is much less than 1, which is usually the case for most practical systems, Eq. (10.29) can be approximated by:

$$A = (1 + \Sigma\, \alpha_i)^{-1} \tag{10.30}$$

Caution is necessary in computing α_i , since Eq. (10.30) applies to the availability of the whole system. Thus, when the units are replaceable as line replaceable units or system replaceable units, the $MTTR_i$ is the mean time required to replace the unit with a good one at the system

maintenance level and is not the mean repair time of the failed removed unit. On the other hand, if failed units are not replaced but are repaired at the system level, $MTTR_i$ is the mean-time-to-repair of the unit, which becomes also the downtime for the system. Thus, when computing the A_s of the units and the availability A_s of the system, all MTTRs must be those repair times that the system experiences as its own downtime. The $MTTR_i$ of the i^{th} unit is thus the system mean repair time when the i^{th} unit fails.

If we compare Eq. (10.30) with Eq. (10.20) in Model A we find that they are identical. The system maintenance time ratio (MTR) is:

$$\alpha = MTTR/MTBF \tag{10.31}$$

But the serial system's MTTR as shown in Section 4 is given by:

$$MTTR = \Sigma\lambda_i (MTTR_i)/\Sigma\lambda_i \tag{10.32}$$

while its MTBF is

$$MTBF = (\Sigma\lambda_i)^{-1}$$

$$= \Sigma\lambda_i (MTTR_i)\Sigma\lambda_i/\Sigma\lambda_i$$

$$= \Sigma\lambda_i (MTTR_i) = \Sigma\alpha_i \tag{10.33}$$

where:

$$\lambda_i = \frac{1}{MTBF_i}$$

In other words, the system MTR is the sum of the unit MTRs. The MTR is actually the average system downtime per system operating hour. Conceptually, it is very similar to the maintenance ratio (MR) defined as maintenance man-hours expended per system operating hour. The difference is that in the MTR one looks only at system downtime in terms of clock hours of system repair, whereas in the MR one looks at all maintenance man-hours expended at all maintenance levels to support system operation.

Eq. (10.30) can be still further simplified if $\sum_{i=1}^{N} \lambda_i/\mu_i < 0.1$

In that case

$$A \approx 1 - \sum_{i=1}^{N} \lambda_i/\mu_i \qquad (10.34)$$

or the system availability is equal to 1 - (the sum of the unit MTRs).

Let us work some examples.

Example 5:

Figure 10.4-4 represents a serial system consisting of 5 statistically independent subsystems, each with the indicated MTBF and MTTR. Find the steady state availability of the system.

Note that for the system, we cannot use any of the simplifying assumptions since, for example, subsystems 3 and 4 have MTRs of 0.2 and 0.1, respectively, which are not << than 1.

Also $\sum_{i=1}^{N} \lambda_i/\mu_i = 0.33$ which is not < 0.1.

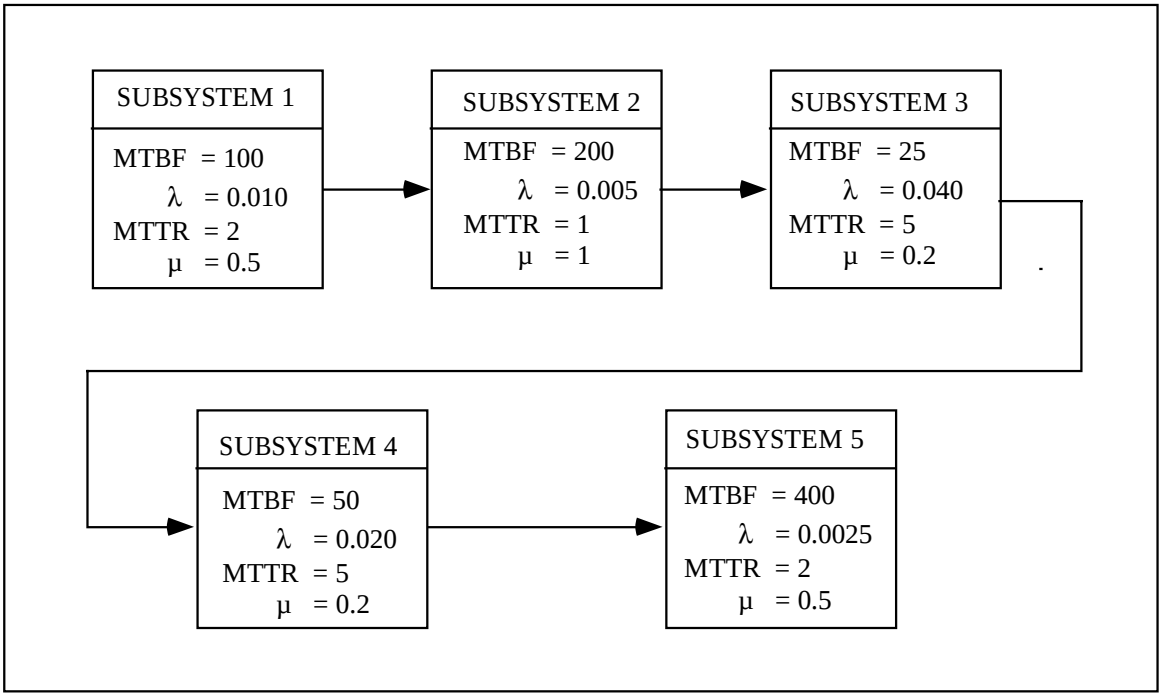


FIGURE 10.4-4: BLOCK DIAGRAM OF A SERIES SYSTEM

Therefore, we must use the basic relationship, Eq. (10.27).

$$A = \prod_{i=1}^{N} \left(\frac{1}{1 + \frac{MTTR_i}{MTBF_i}} \right)$$

$$= \left(\frac{1}{1 + 2/100}\right)\left(\frac{1}{1 + 1/200}\right)\left(\frac{1}{1 + 5/25}\right)\left(\frac{1}{1 + 5/50}\right)\left(\frac{1}{1 + 2/400}\right)$$

$$= (0.98039)\ (0.99502)\ (0.83333)\ (0.90909)\ (0.99502) = 0.73534$$

Example 6:

Now let us look at a similar series system, consisting of 5 statistically independent subsystems having the following MTBFs and MTTRs, as shown in the table below.

Subsystem	MTBF	MTTR	α	A
1	100	0.5	0.005	0.995
2	200	1	0.005	0.995
3	300	0.75	0.0025	0.9975
4	350	1.5	0.0043	0.9957
5	500	2	0.004	0.996

In this case, each α_i is << than 1 and $\sum_{i=1}^{5} \alpha_i < .1$, so that we can use the simplified Eq. (10.34).

$$A \approx 1 - \sum_{i=1}^{5} \lambda_i/\mu_i = 1 - 0.0208 = 0.9792$$

Of course, the power and speed of modern hand-held calculators and personal computers tend to negate the benefits of the simplifying assumptions.

10.4.1.4 Model D - Redundant Systems

(See Section 7.5 for a more detailed description of the mathematical models used to calculate the reliability of systems incorporating some form of redundancy). In this model, the availability of some redundant systems is considered. First we deal with two equal, independent units in a parallel redundant arrangement with each unit being separately repairable or replaceable while the other continues operating. Thus, the system is "up" if both or any one of the two units

operates. (See Section 7.5 for a more detailed description of the mathematical models used to calculate the reliability of systems incorporating some form of redundancy).

If we define the unavailability U of a unit as

$$U = 1 - A = MTTR/(MTBF + MTTR) \quad (10.35)$$

then the probability that the system is unavailable is the probability that both units are down at the same time, which is

$$U_{system} = U^2 \quad (10.36)$$

and system availability is

$$A_{system} = 1 - U^2 \quad (10.37)$$

Further, using the binomial expansion

$$(A + U)^2 = A^2 + 2AU + U^2 = 1 \quad (10.38)$$

we find that we may write Eq. (10.38) also in the form

$$A_{system} = A^2 + 2AU \quad (10.39)$$

which gives us the probability A^2 that both units are operating at any point in time and the probability 2AU that only one unit is working. Over a period of time T, the system will on the average be operating for a time TA^2 with both units up, while for 2TAU only one unit will be up. If the performance of the system is P_1 when both units are up and P_2 when only one unit is up, the system output or effectiveness, SE, over T^2 is expected to be

$$SE = P_1 TA^2 + 2P_2 TAU \quad (10.40)$$

Assume a ship has two engines which are subject to on-board repair when they fail. When both engines work, the ship speed is 30 nmi/hour, and when only one engine works it is 20 nmi/hour. Let an engine MTBF be 90 hr. and let its MTTR be 10 hr., so that the availability of an engine is A = 0.9 and its unavailability is U = 0.1. Over a 24-hour cruise the ship will be expected to travel on the average

$$SE = 30 \cdot 24 \cdot .81 + 2 \cdot 20 \cdot 24 \cdot 0.9 \cdot 0.1 = 583.2 + 86.4 = 669.6 \text{ nmi.}$$

The expected time for the ship to be found idle with both engines out for a 24-hour cruise is:

$$T_{idle} = 24U^2 = 24(0.01) = 0.24 \text{ hr} \tag{10.41}$$

For three units in parallel we get

$$(A + U)^3 = A^3 + 3A^2 U + 3AU^2 + U^3 = 1 \tag{10.42}$$

If the system goes down only if all three units are down, system availability is:

$$A_{system} = A^3 + 3A^2 U + 3AU^2 = 1 - U^3 \tag{10.43}$$

but if at least two units are needed for system operation since a single unit is not sufficient, system availability becomes

$$A_{system} = A^3 + 3A^2 U \tag{10.44}$$

In general, for a system with n equal, redundant units, we expand the binomial term

$$(A + U)^n = 1, \text{ or}$$

$$A^n + \left(nA^{n-1}U\right) + \left(\frac{n(n-1)}{2!} A^{n-2} U^2\right) + \left(\frac{n(n-1)(n-2)}{3!} A^{n-3} U^3\right) + \ldots + U^n = 1 \tag{10.45}$$

which yields the probabilities of being in any one of the possible states. Then, by adding the probabilities of the acceptable states, we obtain the availability of the system. As stated earlier, the units must be independent of each other, both in terms of their failures and in terms of their repairs or replacements, with no queuing up for repair.

Reference [9] contains, throughout the text, extensive tabulations of availability and related measures of multiple parallel and standby redundant systems for cases of unrestricted as well as restricted repair when failed redundant units must queue up and wait until their turn comes to get repaired.

Returning briefly to Eq. (10.36), when the two redundant units are not equal but have unavailabilities $U_1 = 1 - A_1$ and $U_2 = 1 - A_2$, system unavailability becomes:

$$U_{system} = U_1 U_2 \tag{10.46}$$

and availability

$$A_{system} = 1 - U_1 U_2 \tag{10.47}$$

Again, we may expand the multinomial

$$(A_1 + U_1)(A_2 + U_2) = A_1A_2 + A_1U_2 + A_2U_1 + U_1U_2 \tag{10.48}$$

and may write system availability in the form

$$A_{system} = A_1A_2 + A_1U_2 + A_2U_1 \tag{10.49}$$

For n unequal units we expand the term

$$\sum_{i=1}^{n} (A_i + U_i) = 1 \tag{10.50}$$

and add together the probabilities of acceptable states and other effectiveness measures, as illustrated in the ship engines example.

This approach is analogous to that shown in Section 5 (k out of n configuration) for reliability.

It can be shown that the limiting expression for an n equipment parallel redundant system reduces to the binomial form if there are as many repairmen as equipments. This is equivalent to treating each equipment as if it had a repairman assigned to it or to saying that a single repairman is assigned to the system but that the probability of a second failure occurring while the first is being repaired is very small. The expression for steady state availability is

$$A\left[\, 1/n \,\right] = 1 - (1 - A)^n \tag{10.51}$$

where n is the number of redundant equipments and l/n indicates that at least 1 equipment must be available for the system to be available.

In general where at least m out of n redundant equipments must be available for the system to be available:

$$A\left[\, m/n \,\right] = \sum_{i=m}^{n} \binom{n}{i} A^i (1 - A)^{n-i}$$

$$= \sum_{i=m}^{n} \frac{n!}{(n-1)!\,i!} \left(\frac{\mu}{\mu+\lambda}\right)^i \left(\frac{\lambda}{\mu+\lambda}\right)^{n-1} \tag{10.52}$$

Table 10.4-1 (Ref. [10]) provides expressions for the instantaneous and steady state availability for 1, 2, and 3 equipments, parallel and standby redundancy, and single and multiple repair maintenance policies.

Single repair means that failed units can be repaired one at a time. If a unit fails, repairs are immediately initiated on it. If more than one unit is down, repairs are initiated on a single unit until it is fully operational; then, repairs are initiated on the second failed unit. For the case of ***multiple repair***, all failed units can have repair work initiated on them as soon as failure occurs, and the work continues until each unit is operational. Also, a repair action on one unit is assumed to be independent of any other unit.

One case not yet addressed is the case of redundant units when repairs cannot be made until complete system failure (all redundant units have failed). The steady state availability can be approximated by (see Ref. [25] for deriving exact expressions):

$$A = \frac{MTTF}{MTTF + MTTR} \tag{10.53}$$

where:

MTTF = mean time to failure for redundant system

and

MTTR = mean time to restore all units in the redundant system

In the case of an n-unit ***parallel*** system

$$MTTF = \sum_{n=1}^{n} \frac{1}{i\lambda} \tag{10.54}$$

and

$$MTTR = \frac{m}{\mu} \tag{10.55}$$

TABLE 10.4-1: AVAILABILITY OF SOME REDUNDANT SYSTEMS BASED ON EXPONENTIAL FAILURE AND REPAIR DISTRIBUTIONS

No. of Equipments	Conditions: Type Redundancy	Conditions: Repair	Instantaneous Availability Model	Definitions of Constants for Instantaneous Availability Model	Steady State Availability: Model	Steady State Availability: A_{system} for $\lambda = 0.01$ $\mu = 0.2$
1	---	---	$A(t) = \frac{\mu}{\mu+\lambda} + \frac{\lambda}{\mu+\lambda} e^{-(\mu+\lambda)t}$	———	$\frac{\mu}{\mu+\lambda}$	0.95
2	Standby	Single	$A(t) = \frac{\mu^2+\mu\lambda}{\mu^2+\mu\lambda+\lambda^2} - \frac{\lambda^2(s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_1 s_2 (s_1 - s_2)}$	$s_1 = -(\lambda+\mu) - \sqrt{\mu\lambda}$ $s_2 = -(\lambda+\mu) + \sqrt{\mu\lambda}$	$\frac{\mu^2+\mu\lambda}{\mu^2+\mu\lambda+\lambda^2}$	0.998
		Multiple	$A(t) = \frac{2\mu^2+2\mu\lambda}{2\mu^2+2\mu\lambda+\lambda^2} - \frac{\lambda^2(s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_1 s_2 (s_1 - s_2)}$	$s_1 = -\frac{1}{2}\left[(2\lambda+3\mu) + \sqrt{\mu^2+4\mu\lambda}\right]$ $s_2 = -\frac{1}{2}\left[(2\lambda+3\mu) - \sqrt{\mu^2+4\mu\lambda}\right]$	$\frac{2\mu^2+2\mu\lambda}{2\mu^2+2\mu\lambda+\lambda^2}$	0.999
	Parallel	Single	$A(t) = \frac{\mu^2+2\mu\lambda}{\mu^2+2\mu\lambda+2\lambda^2} - \frac{2\lambda^2(s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_1 s_2 (s_1 - s_2)}$	$s_1 = -\frac{1}{2}\left[(3\lambda+2\mu) + \sqrt{\lambda^2+4\mu\lambda}\right]$ $s_2 = -\frac{1}{2}\left[(3\lambda+2\mu) - \sqrt{\lambda^2+4\mu\lambda}\right]$	$\frac{\mu^2+2\mu\lambda}{\mu^2+2\mu\lambda+2\lambda^2}$	0.996
		Multiple	$A(t) = \frac{\mu^2+2\mu\lambda}{\mu^2+2\mu\lambda+\lambda^2} - \frac{2\lambda^2(s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_1 s_2 (s_1 - s_2)}$	$s_1 = -2(\mu+\lambda)$ $s_2 = -(\mu+\lambda)$	$\frac{\mu^2+2\mu\lambda}{\mu^2+2\mu\lambda+\lambda^2}$	0.998
3	Standby	Single	$A(t) = \frac{\mu^3+\mu^2\lambda+\mu\lambda^2}{\mu^3+\mu^2\lambda+\mu\lambda^2+\lambda^3} + \frac{\lambda^3[s_2 s_3(s_2-s_3)e^{s_1 t} - s_1 s_3(s_1-s_3)e^{s_2 t} + s_1 s_2(s_1-s_2)e^{s_3 t}]}{s_1 s_2 s_3 (s_1-s_2)(s_1-s_3)(s_2-s_3)}$	s_1, s_2, and s_3 correspond to the three roots of $s^3 + s^2(3\lambda+3\mu) + s(3\lambda^2+4\mu\lambda+3\mu^2) + (\lambda^3+\mu\lambda^2+\lambda\mu^2+\mu^3)$	$\frac{\mu^3+\mu^2\lambda+\lambda^2\mu}{\mu^3+\mu^2\lambda+\lambda^2\mu+\lambda^3}$	0.9999
		Multiple	$A(t) = \frac{6\mu^3+6\mu^2\lambda+3\mu\lambda^2}{6\mu^3+6\mu^2\lambda+3\mu\lambda^2+\lambda^3} + \frac{\lambda^3[s_2 s_3(s_2-s_3)e^{s_1 t} - s_1 s_3(s_1-s_3)e^{s_2 t} + s_1 s_2(s_1-s_2)e^{s_3 t}]}{s_1 s_2 s_3 (s_1-s_2)(s_1-s_3)(s_2-s_3)}$	s_1, s_2, and s_3 correspond to the three roots of $s^3 + s^2(3\lambda+3\mu) + s(3\lambda^2+9\mu\lambda+11\mu^2) + (\lambda^3+3\mu\lambda^2+6\mu^2\lambda+6\mu^3)$	$\frac{6\mu^3+6\mu^2\lambda+3\mu\lambda^2}{6\mu^3+6\mu^2\lambda+3\mu\lambda^2+\lambda^3}$	0.99998
	Parallel	Single	$A(t) = \frac{\mu^3+3\mu^2\lambda+6\mu\lambda^2}{\mu^3+3\mu^2\lambda+6\mu\lambda^2+6\lambda^3} + \frac{6\lambda^3[s_2 s_3(s_2-s_3)e^{s_1 t} - s_1 s_3(s_1-s_3)e^{s_2 t} + s_1 s_2(s_1-s_2)e^{s_3 t}]}{s_1 s_2 s_3 (s_1-s_2)(s_1-s_3)(s_2-s_3)}$	s_1, s_2, and s_3 correspond to the three roots of $s^3 + s^2(6\lambda+3\mu) + s(11\lambda^2+9\mu\lambda+3\mu^2) + (6\lambda^3+6\mu\lambda^2+3\mu^2+\mu^3)$	$\frac{\mu^3+3\mu^2\lambda+6\mu\lambda^2}{\mu^3+3\mu^2\lambda+6\mu\lambda^2+6\lambda^3}$	0.9993
		Multiple	$A(t) = \frac{\mu^3+3\mu^2\lambda+3\mu\lambda^2}{(\mu+\lambda)^3} + \frac{6\lambda^3[s_2 s_3(s_2-s_3)e^{s_1 t} - s_1 s_3(s_1-s_3)e^{s_2 t} + s_1 s_2(s_1-s_2)e^{s_3 t}]}{s_1 s_2 s_3 (s_1-s_2)(s_1-s_3)(s_2-s_3)}$	s_1, s_2, and s_3 correspond to the three roots of $s^3 + s^2(6\lambda+6\mu) + s[11(\mu+\lambda)^2] + 6(\mu+\lambda)^3$	$\frac{\mu^3+3\mu^2\lambda+3\mu\lambda^2}{\mu^3+3\mu^2\lambda+3\mu\lambda^2+\lambda^3}$	0.9999

NOTES:
1. A(t) is the probability of a system being available at time t: A(t) is a function of μ and λ the repair and failure rates. For all functions, the probability of a system being available at time zero is unity. The units of μ and λ must be the same as for t.
2. Instantaneous availability. The probability that the system will be available at any instant in time.
3. Mission availability. Expected availability for a given mission period. This value can be derived from the general model by computing the average value of A(t) for the mission period. Mathematically, this is $A_m = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} A(t)dt$. Usually t_1 is considered as zero.
4. Steady state availability. The portion of up-time expected for continuous operation.

where:

m = 1, for the multiple repairs case

and

m = n, for the single repair case, or

$$A\ (1/n) = \frac{\sum_{i=1}^{n} \frac{1}{i\lambda}}{\sum_{i=1}^{n} \frac{1}{i\lambda} + \frac{m}{\mu}} \tag{10.56}$$

In the case of an n-unit ***standby*** system with one active and n-1 standby units

$$MTTF = \frac{n}{\lambda} \tag{10.57}$$

and

$$MTTR = \frac{m}{\lambda} \tag{10.58}$$

where:

m = 1, for the multiple repairs case

and

m = n, for the single repair case.

Then

$$A = \frac{n/\lambda}{n/\lambda + m/\lambda} \tag{10.59}$$

Following are some examples utilizing the concepts presented in this section.

Example 7:

In the case of a 2-unit parallel system with $\lambda = 0.01$ fr/hr and $\mu = 1.0$ rp/hr, if the system does not undergo repairs until both units fail, the system's steady-state availability is by Eq. (10.56).

$$A[1/2] = \frac{\sum_{n=1}^{2} \frac{1}{n\lambda}}{\sum_{n=1}^{2} \frac{1}{n\lambda} + \frac{m}{\mu}}$$

With single repair (Case 1)

$$A[1/2] = \frac{\frac{1}{\lambda} + \frac{1}{2\lambda}}{\frac{1}{\lambda} + \frac{1}{2\lambda} + \frac{2}{\mu}} = \frac{\frac{1}{0.01} + \frac{1}{2(0.01)}}{\frac{1}{0.01} + \frac{1}{2(0.01)} + 2} = 150/152 = 0.9868$$

With multiple repairs (Case 2)

$$A(1/2) = \frac{\frac{1}{\lambda} + \frac{1}{2\lambda}}{\frac{1}{\lambda} + \frac{1}{2\lambda} + \frac{1}{\mu}} \quad \text{or} \quad A(1/2) = \frac{\frac{1}{0.01} + \frac{1}{2(0.01)}}{\frac{1}{0.01} + \frac{1}{2(0.01)} + \frac{1}{1}}$$

$$A(1/2) = 0.9934$$

If repairs are initiated each time a unit fails, with multiple repairs when both units fail (Case 3) then from Table 10.4-1.

$$A(1/2) = \frac{\mu^2 + 2\lambda\mu}{\mu^2 + 2\mu\lambda + \lambda^2} \quad \text{or} \quad A(1/2) = \frac{(1)^2 + 2(0.01)(1)}{(1)^2 + 2(1)(0.01) + (0.01)^2}$$

and

$$A(1/2) = 0.9999$$

Looking at the three cases of this example

	Availability	Average Downtime in 10,000 hours
Case 1	0.9868	132 hrs.
Case 2	0.9934	66 hrs.
Case 3	0.9999	1 hr.

We can see that the maintenance philosophy plays a significant role. For example, Cases 1 and 2 may not be acceptable for a crucial system such as a ballistic missile early warning system.

Example 8:

We have three redundant equipments, each with an availability of 0.9. What is the availability of the configuration if two of the three equipments must be available at anytime?

(a) From Eq. (10.45)

$$A^3 + 3A^2U + 3AU^2 + U^3 = 1$$

$$A^3 + 3A^2U = (0.9)^3 + 3\,(0.9)^2(0.1)$$

$$= 0.729 + 0.243 = 0.972$$

(b) From Eq. (10.52)

$$A(2/3) = \frac{3!}{(3-2)!\,2!}\,(0.9)^2(0.1)^{3-2} + \frac{3!}{(3-3)!3!}\,(0.9)^3(0.1)^{3-3}$$

$$= 3(0.9)^2(0.1) + (0.9)^3 = 0.972$$

Example 9:

Given three standby equipments with multiple repair capability, the MTBF of each equipment is 1000 hours and the repair rate is 0.02/hr. What is the expected steady state availability (A_{ss})?

From Table 10.4-1, we see that the appropriate formula is

$$A_{ss} = \frac{6\mu^3 + 6\mu^2\lambda + 3\mu\lambda^2}{6\mu^3 + 6\mu^2\lambda + 3\mu\lambda^2 + \lambda^3}$$

$$\lambda = 1/1000 = 0.001/hr$$

$$\mu = 0.02/hr$$

Substituting these values

$$A_{ss} = \frac{6(0.02)^3 + 6(0.02)^2(0.001) + 3(0.02)(0.001)^2}{6(0.02)^3 + 6(0.02)^2(0.001) + 3(0.02)(0.001)^2 + (0.001)^3}$$

$$= \frac{6(0.000008) + 6(0.0004)(0.001) + (0.06)(0.000001)}{6(0.000008) + 6(0.0004)(0.001) + (0.06)(0.000001) + (0.001)^3}$$

$$= \frac{0.000048000 + 0.00000240 + 0.00000006}{0.000048000 + 0.00000240 + 0.000000060 + 000000001}$$

$$= \frac{5.046 \times 10^{-5}}{5.0461 \times 10^{-5}} = 0.99998$$

Example 10:

Given two standby equipments in an early warning ground radar system. The equipments are operated in parallel and have a single repair capability. The MTBF of each equipment is 100 hours and the repair rate is 2/hr. What is the expected steady state availability?

From Table 10.4-1, the appropriate equation is:

$$A_{ss} = \frac{\mu^2 + \mu\lambda}{\mu^2 + \mu\lambda + \lambda^2} = \frac{(2)^2 + 2(0.01)}{(2)^2 + 2(0.01) + (0.01)^2} = \frac{4.02}{4.0201} = 0.999975$$

Example 11:

Let us return to the example of the previous section, Figure 10.4-4, in which we had a series system consisting of five subsystems with the following R&M parameters:

Subsystem	λ	μ	A (previously calculated)
1	0.01	0.5	0.98039
2	0.005	1	0.99502
3	0.04	0.2	0.83333
4	0.02	0.2	0.90909
5	0.0025	0.5	0.99502

It was previously found that the availability of this system was $\prod_{i=1}^{5} A_i = 0.73534$

Suppose that we would like to raise the system availability to 0.95 by using redundant parallel subsystems with multiple repair for subsystems 3 and 4 (the two with lowest availability). How many redundant subsystems would we need for each subsystem?

We have the situation

$$A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 = 0.95$$

$$A_3 \cdot A_4 = \frac{0.95}{A_1 A_2 A_5} = \frac{0.95}{(0.98039)(0.99502)(0.99502)} = \frac{0.95}{0.97065} \approx 0.98$$

This means that the product of the improved availabilities ($A_3 A_4$) of subsystems 3 and 4 must be approximately 0.98. As a first cut, let us assume equal availability for improved subsystems 3 and 4. This means that each must have an availability of 0.99 for their product to be 0.98.

Eq. (10.51) is the general expression for improvement in availability through redundancy

$$A(1/n) = 1 - (1 - A)^n$$

where A(1/n) is the improved availability with n redundant units. Let us call this A'. Then,

$$A' = 1 - (1 - A)^n$$

and

$$1 - A' = (1 - A)^n$$

Taking the natural logarithm of both sides of the equation

$$ln\,(1 - A') = n\,ln\,(1 - A)$$

$$n = \frac{ln(1-A')}{ln(1-A)} \qquad (10.60)$$

which is a general expression that can be used to determine the number of redundant subsystems required to achieve a desired subsystem availability (A').

Let us look at improved subsystem 3:

$$A' = 0.99$$

$$A = 0.83333$$

$$n = \frac{ln(1-0.99)}{ln(1-0.8333)} = \frac{ln(0.01)}{ln(0.16667)} = \frac{-4.605}{-1.79}$$

$$= 2.57, \text{ which is rounded up to 3 redundant subsystems (total).}$$

Similarly for subsystem 4:

$$n = \frac{ln(1-0.99)}{ln(1-0.90909)} = \frac{ln(0.01)}{ln(0.09091)} = \frac{-4.605}{-2.397}$$

$$= 1.92, \text{ which is rounded up to 2 redundant subsystems}$$

Thus, in order for the system availability to be raised to 0.95, we need 3 parallel redundant Subsystems 3, and 2 parallel redundant Subsystems 4.

Note that we have not discussed the optimum allocation of failure and repair rates to achieve a given availability; this will be done later in this section.

10.4.1.5 Model E - R&M Parameters Not Defined in Terms of Time

A very different situation in availability modeling is encountered when system "uptime" is not measured in hours of operation or any time parameter but rather in terms of number of rounds fired, miles traveled, actuations or cycles performed, etc. The reliability parameter is then no longer expressed in terms of MTBF but rather in mean-rounds-between-failures (MRBF), mean-miles-between-failures (MMBF), mean-cycles-between-failures (MCBF), etc. The failure rate then also is expressed in number of failures per round, per mile, or per cycle rather than number of failures per operating hour.

For straightforward reliability calculations this poses no problem since the same reliability equations apply as in the time domain, except that the variable time, t, in hours is replaced by the variable number of rounds, number of miles, etc. We may then calculate the reliability of such systems for one, ten, one hundred, or any number of rounds fired or miles traveled, as we wish. The maintainability calculations remain as before, since downtime will always be measured in terms of time and the parameter of main interest remains the MTTR.

However, when it comes to availability, which usually combines two time parameters (i.e., the MTBF and the MTTR into a probability of the system being up at some time, t), a difficult problem arises when the time, t, is replaced by rounds or miles, since the correlation between time and rounds or time and miles is quite variable.

An equation for the steady-state availability of machine guns is given in Reference [11]. This equation is based on a mission profile that at discrete times, t_1, t_2, t_3, etc., requires the firing of N_1, N_2, N_3, etc., bursts of rounds. When the gun fails during a firing, for example at time t, it fires only f rounds instead of N_3 rounds and must undergo repair during which time it is not available to fire; for example, it fails to fire a required N_4 rounds at t_4, and a further N_5 rounds at t_5 before becoming again available (see Figure 10.4-5). Its system availability, A, based on the rounds not fired during repair may be expressed, for the described history, as:

$$A = (N_1 + N_2 + f)/(N_1 + N_2 + N_3 + N_4 + N_5) \tag{10.61}$$

Each sequence of rounds fired followed by rounds missed (not fired) constitutes a renewal process in terms of rounds fired, as shown in Figure 10.4-6, where the gun fails after firing x rounds, fails to fire $\gamma(x)$ rounds in the burst of rounds during which it failed and also misses firing the required bursts of rounds while in repair for an MTTR = M. Assume that the requirements for firing bursts of rounds arrives at random according to a Poisson process with rate r and the average number of rounds per burst is N, then the limiting availability of the gun may be expressed as:

$$A = MRBF/(MRBF + N + \gamma MN) \quad (10.62)$$

where MRBF is the mean number of rounds between failure. The derivation of this formula, developed by R.E. Barlow, is contained in the Appendix of Reference [11]. To calculate A from Eq. (10.62) one must know the MRBF and MTTR of the gun, the average rounds N fired per burst, and the rate γ at which requirements for firing bursts of rounds arrive.

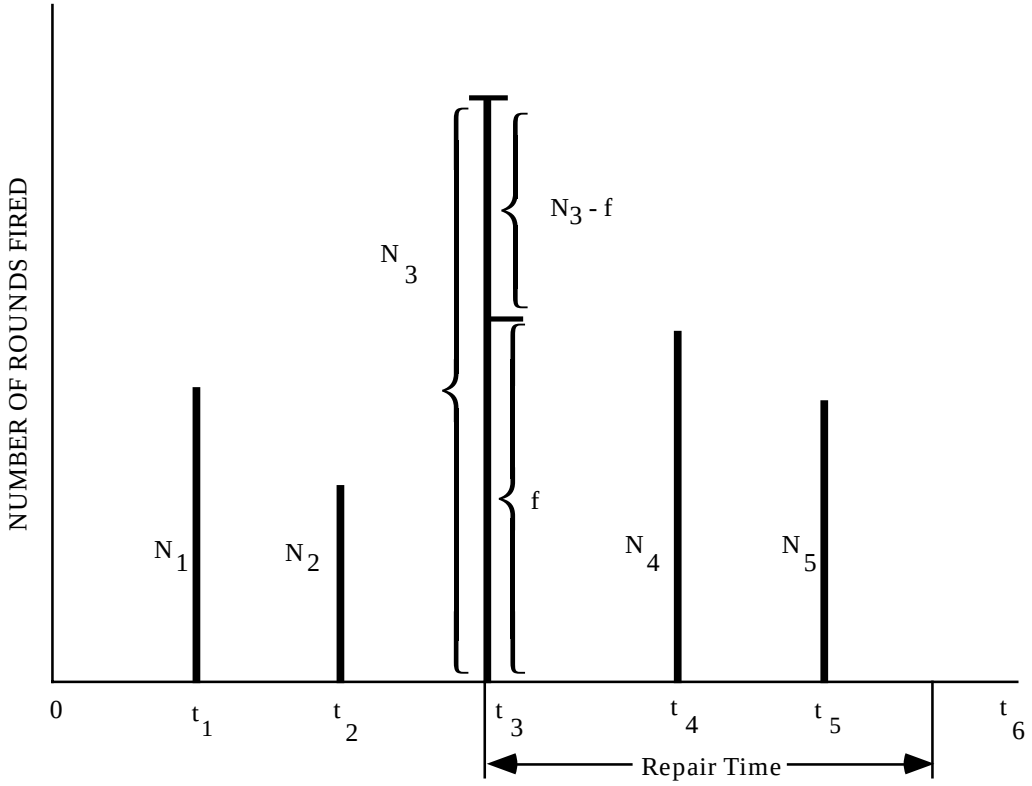


FIGURE 10.4-5: HYPOTHETICAL HISTORY OF MACHINE GUN USAGE

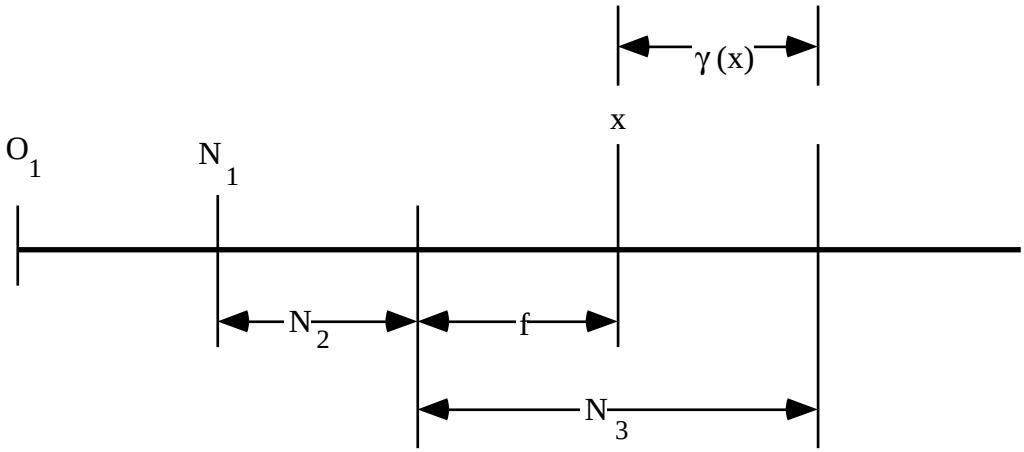


FIGURE 10.4-6: RENEWAL PROCESS IN TERMS OF ROUNDS FIRED

Similar availability equations can be developed for other types of weapons and also for vehicles where the renewal process is in terms of miles traveled. Other approaches to calculating the availability of guns as well as vehicles are found in Reference [12] and are based on calculating from historical field data the maintenance ratios and, via regression analysis, the maintenance time ratios (called the "maintenance clock hour index") that are in turn used in the conventional time based equation of inherent, achieved, and operational availability.

For example, consider a machine gun system in a tank on which historical data are available, showing that 0.014 corrective maintenance man-hours are expended per round fired and that per year 4800 rounds are fired while the vehicle travels for 240 hr per yr. The maintenance ratio (MR) for the gun system is then computed as (Ref. [12], pp. 36-38).

$$MR_{Gun} = \frac{MMH}{Round} \cdot \frac{\text{Number of Rounds Fired per Annum}}{\text{Vehicle Operating Hours per Annum}}$$

$$= 0.014 \cdot (4800/240) = 0.28 \qquad (10.63)$$

The dimensions for 0.28 are gun system maintenance man-hours per vehicle operating hour. According to this example, the corrective maintenance time ratio, α (sometimes called the maintenance clock hour index, Ω), is, given by:

$$\alpha_{Gun} = 0.628(0.28)^{0.952} = 0.187 \qquad (10.64)$$

The numbers 0.628 and 0.952 are the intercept and the regression coefficients, respectively, obtained by regression analysis as developed in Reference [12], p. 18, Table 1. The dimension for α_{Gun} is gun system downtime per vehicle operating hour. The inherent availability of the gun system is then, according to the conventional time equation, Eq. (10.20).

$$A_i = (1 + \alpha Gun)^{-1} = (1.187)^{-1} = 0.842 \quad (10.65)$$

This may be interpreted as the gun system being available for 84.2% of the vehicle operating time. Caution is required in using this approach for weapon availability calculations, since in the case where the vehicle would have to be stationary and the gun would still fire rounds, MR and α would become infinitely large and the inherent availability of the gun system would become zero.

10.4.2 Mission Reliability and Dependability Models

Although availability is a simple and appealing concept at first glance, it is a point concept, i.e., it refers to the probability of a system being operable at a random point in time. However, the ability of the system to continue to perform reliably for the duration of the desired operating (mission) period is often more significant. Operation over the desired period of time depends, then, on clearly defining system operating profiles. If the system has a number of operating modes, then the operating profile for each mode can be considered.

The term mission reliability has been used to denote the system reliability requirement for a particular interval of time. Thus, if the system has a constant failure rate region so that its reliability R can be expressed as:

$$R = \exp(-\lambda t) \quad (10.66)$$

where:

λ = failure rate = 1/MTBF
t = time for mission

then mission reliability R_M for a mission duration of T is expressed as:

$$R_M = \exp(-\lambda T) \quad (10.67)$$

This reliability assessment, however, is conditional upon the system being operable at the beginning of its mission or its (point) availability.

In order to combine these two concepts, a simplified system effectiveness model may be used where the system effectiveness may be construed simply as the product of the probabilities that the system is operationally ready and that it is mission reliable.

If A is the mean availability of a system at any point in time t_o when we want to use the system and if R_M is the system reliability during mission time T, then system effectiveness E, not including performance, may be defined as:

$$E = AR_M \tag{10.68}$$

Thus, A is a weighting factor, and E represents an assessment of system ability to operate without failure during a randomly chosen mission period.

One concept of dependability used by the Navy (Ref. [13]) takes into account the fact that for some systems a failure which occurs during an operating period t_1 may be acceptable if the failure can be corrected in a time t_2 and the system continues to complete its mission. According to this concept, dependability may be represented by:

$$D = R_M + (1 - R_M)M_o \tag{10.69}$$

where:

- D = system dependability - or the probability that the mission will be successfully completed within the mission time t_1, providing a downtime per failure not exceeding a given time t_2 will not adversely affect the overall mission
- R_M = mission reliability - or the probability that the system will operate without failure for the mission time t_1
- M_o = operational maintainability - or the probability that when a failure occurs, it will be repaired in a time not exceeding the allowable downtime t_2
- t_2 = specified period of time within which the system must be returned to operation

For this model, the exponential approximation of the lognormal maintainability function is used, or

$$M_o = \left(1 - e^{-\mu t_2}\right) \tag{10.70}$$

Then, the system effectiveness is:

$$E = AD = A\left[R_M + (1 - R_M)\, M_o\right] \tag{10.71}$$

In the case where no maintenance is allowed during the mission ($t_2 = 0$ or $M_o = 0$), as in the case of a missile, then this reduces to Eq. (10.68).

$$E = AD = AR_M$$

This concept of dependability is compatible with the WSEIAC model and indeed can be taken into account in the dependability state transition matrices.

Let us examine an airborne system with the following parameters and requirements:

$\lambda = 0.028$ failures/hr

$\mu = 1$ repair/hr

Mission time (T) = 8 hours

t_a = 30 minutes to repair a failure during a mission

Thus,

$$A = \frac{\mu}{\mu + \lambda} = \frac{1}{1 + 0.028} = .973 \text{ at the start of the mission}$$

$$R_M = e^{-\lambda T} = e^{-(0.028)(8)} = 0.8 \text{ (mission reliability)}$$

$$M_o = 1 - e^{-\mu t_a} = 1 - e^{-(1)(0.5)} = 0.4$$ (probability of repairing failure during mission within $\frac{1}{2}$ hour)

$$\therefore E = A\left[R_M + (1 - R_M) M_o \right]$$

$$= 0.973\left[0.8 + (1 - 0.8)(0.4) \right]$$

$$= 0.973\left[0.8 + 0.08 \right] = 0.86$$

10.4.3 Operational Readiness Models

Availability, defined as the uptime ratio, is not always a sufficient measure to describe the ability of a system to be committed to a mission at any arbitrary time. In many practical military operations, the concept of operational readiness serves this purpose better. We here define

operational readiness as the probability that a system is in an operable condition, i.e., ready to be committed to perform a mission when demands for its use arise. The difference as well as the similarity between availability and operational readiness will become clear by comparing the models developed subsequently with the availability models discussed in the preceding section.

In the development of operational readiness models, one has to consider the usage and the maintenance of the system, i.e., its operating, idle, and repair times. When a call arrives for the system to engage in a mission, the system at such time may be in a state of perfect repair and ready to operate immediately. But it may also be in need of maintenance and not ready. Its state when called upon to operate depends on the preceding usage of the system, i.e., on its preceding mission, in what condition it returned from that mission, and how much time has elapsed since it completed the last mission. Many models can be developed for specific cases, and some are discussed in the following paragraphs.

10.4.3.1 Model A - Based Upon Probability of Failure During Previous Mission and Probability of Repair Before Next Mission Demand

In this model, the assumption is made that if no failures needing repair occurred in the preceding mission, the system is immediately ready to be used again; and, if such failures did occur, the system will be ready for the next mission only if its maintenance time is shorter than the time by which the demand for its use arises. The operational readiness P_{OR} may then be expressed as:

$$P_{OR} = R(t) + Q(t) \bullet P(t_m < t_d) \tag{10.72}$$

where:

$R(t)$ = probability of no failures in the preceding mission

$Q(t)$ = probability of one or more failures in the preceding mission

t = mission duration

$P(t_m < t_d)$ = probability that if failures occur, the system maintenance time, t_m, is shorter than the time, t_d, at which the next demand or call for mission engagement arrives

The calculations of $R(t)$ and $Q(t) = 1 - R(t)$ are comparatively simple using standard reliability equations; however, all possible types of failures that need fixing upon return in order to restore in full the system reliability and combat capability must be considered, including any failures in redundant configurations.

As for $P(t_m < t_d)$, one needs to know the probability distributions of the system maintenance time and of call arrivals. Denoting by $f(t_m)$ the probability density function of maintenance time and by $g(t_d)$, the probability density function of time to the arrival of the next call, counted from the instant the system returned from the preceding mission in a state requiring repair, the probability that the system will be restored to its full operational capability before the next call arrives is:

$$P(t_m < t_d) = \int_{t_m}^{\infty} f(t_m) \left[\int_{t_d=t_m}^{\infty} g(t_d)dt_d \right] dt_m \quad (10.73)$$

The integral in the square brackets on the right side of the equation is the probability that the call arrives at t_d after a variable time t_m. When this is multiplied by the density function $f(t_m)$ of the duration of maintenance times and integrated over all possible values of t_m, we get $P(t_m < t_d)$.

Now assume that maintenance time t_m and time to next call arrival t_d are exponentially distributed, with M_1 being the mean time to maintain the system and M_2 the mean time to next call arrival. The probability density functions are thus:

$$f(t_m) = \left[\exp(-t_m / M_1)\right]/M_1 \quad (10.74)$$

$$f(t_d) = \left[\exp(-t_d / M_2)\right]/M_2 \quad (10.75)$$

We then obtain

$$P(t_m < t_d) = \int_0^{\infty} M_1^{-1} \exp(-t_m/M_1) \cdot \left[\int_{t_m}^{\infty} M_2^{-1} \exp(-t_d/M_2)dt_d \right] dt_m$$

$$= \int_0^{\infty} (-M_1^{-1}) \exp\left[-(1/M_1 + 1/M_2)t_m \right] dt_m$$

$$= M_2/(M_1 + M_2) \quad (10.76)$$

In this exponential case, system operational readiness becomes

$$P_{OR} = R(t) + Q(t) \cdot \left[M_2 / (M_1 + M_2) \right] \quad (10.77)$$

As a numerical example let us look at a system with a probability of R = 0.8 of returning from a mission of t = 1 hr duration without requiring repair and therefore had a probability of Q = 0.2 that it will require repair. If system mean maintenance time is M_1 = 1 hr and the mean time to next call arrival is M_2 = 2 hr., the operational readiness of the system becomes

$$P = 0.8 + 0.2\,(2/3) = 0.933$$

Comparing this result with the conventional steady-state availability concept and assuming that the system has a mean maintenance time of M_1 = 1 hr and a mean time to failure of M_2 = 5 hr (roughly corresponding to the exponential case of R = 0.8 for a one-hour mission), we obtain a system availability of:

$$A = M_2/(M_1 + M_2) = 5/6 = 0.833$$

which is a result quite different from P_{OR} = 0.933.

10.4.3.2 Model B - Same As Model A Except Mission Duration Time, t is Probabilistic

The operational readiness model of Eq. (10.72) can be extended to the case when mission duration time t is not the same for each mission but is distributed with a density q(t). We then get

$$P_{OR} = \int_0^\infty R(t)q(t)dt + P(t_m < t_d) \int_0^\infty Q(t)q(t)dt \qquad (10.78)$$

Since the integrals in Eq. (10.78) are fixed numbers, we may write:

$$R = \int_0^\infty R(t)q(t)dt, \quad \text{and}$$

$$Q = \int_0^\infty Q(t)q(t)dt \qquad (10.79)$$

and using the symbol P for $P(t_m < t_d)$, i.e., $P = P(t_m < t_d)$, Eq. (10.78) may be written in the form:

$$P_{OR} = R + QP \qquad (10.80)$$

In this equation R is the probability that the system returns without failures from the last mission;

Q = 1 - R is the probability that one or more failures developed in the last mission; and P is the probability that the system will be repaired before the next call arrives if it developed failures. The mission times are variable here with density q(t).

10.4.3.3 Model C - Similar To Model A But Includes Checkout Equipment Detectability

The operational readiness of the system at time t_a is given by:

$$P_{OR}(t_a) = R(t_m) + \left[kM(t_r)\right] \bullet \left[1 - R(t_m)\right] \quad (10.81)$$

where:

$P_{OR}(t_a)$ = probability of system being available for turnaround time, e.g., t_a of 30 minutes, following completion of preceding mission or initial receipt of alert

$R(t_m)$ = probability that the system will survive the specified mission of duration t_m without failure

t_r = specified turnaround time, or maximum downtime for repair required of the system

k = probability that if a system failure occurs it will be detected during the mission or during system checkout following the mission

$M(t_r)$ = probability that a detected system failure can be repaired in time t_r to restore *the* system to operational status

Thus, when mission reliability, mission duration, availability, and turnaround time are specified for the system, the detectability-times-maintainability function for the system is constrained to pass through or exceed the point given by:

$$kM(t_r) \geq \frac{P_{OR}(t_a) - R(t_m)}{\left[1 - R(t_m)\right]}$$

Consider, for example, the following specified operational characteristics for a new weapons system:

Mission Reliability, $R(t_m) = 0.80$ for t_m of 8 hours

Operational Readiness $P_{OR}(t_a) = 0.95$ for turnaround time, t_a of 30 minutes, following completion of preceding mission or initial receipt of alert.

From the requirements, the required detectability-maintainability product (kM) is derived as follows:

$$kM(30) = \frac{P_{OR}(30) - R(8)}{1 - R(8)} = \frac{0.95 - 0.8}{1 - 0.8} = 0.75$$

Therefore, kM(30) = 0.75 is the joint probability, given that a system failure has occurred, that the failure will be detected (either during the mission or during post mission checkout) and will be repaired within 30 minutes following completion of the mission.

Assume that k is to be 0.9, i.e., built-in test equipment is to be incorporated to detect at least 90% of the system failures and provide go/no-go failure indication.

Then, the maintainability requirement is:

$$M(30) = \frac{0.75}{k} = \frac{0.75}{0.9} \approx 0.83$$

which means that 83% of all system repair actions detected during the mission or during post mission checkout must be completed within 30 minutes.

Using the exponential approximation, maintainability as a function of repair time is expressed as the probability of repair in time t_r:

$$M(t_r) = 1 - e^{-\mu t_r} = 1 - e^{-t_r / \overline{M}_{ct}} \qquad (10.82)$$

where:

$\overline{M}_{ct}$ = MTTR

μ = repair rate, $1/\overline{M}_{ct}$

t_r = repair time for which M(t) is to be estimated

The required mean time to repair ($\overline{M}_{ct}$) is found from Eq. (10.82) by taking the natural log of both sides:

$$\overline{M}_{ct} = - \frac{t_r}{ln\left[1 - M(t_r)\right]}$$

Substituting t_r = 30 minutes, and $M(t_r)$, which we previously found to be 0.83,

$$\overline{M}_{ct} = -\frac{30}{ln\,(0.17)} = \frac{-30}{-1.77} \approx 17 \text{ minutes}$$

And from $M(t_{max}) = 0.95$ we find the maximum time for repair of 95% of detected system failures ($M_{max_{ct}}$) as follows:

$$M(t_{max}) = 0.95 = 1 - e^{-M_{max_{ct}}/\overline{M}_{ct}}$$

$$M_{max_{ct}} = -\overline{M}_{ct}\; ln\,(1 - 0.95)$$

$$= -(17)(-3) = 51 \text{ minutes}$$

Thus, these requirements could be established as design requirements in a system development specification.

Detectability Factor, k = 0.90

Mean Time To Repair, $\overline{M}_{ct}$ = 17 minutes

Maximum Time To Repair, $M_{max_{ct}}$ = 51 minutes

10.4.3.4 Model D - For a Population of N Systems

Let N be the total population of systems, e.g., squadron of aircraft. The service facility itself shall be considered as having k channels, each servicing systems at a mean rate μ. The analysis is based on an assumed Poisson distribution of arrivals and on a mean service time which is assumed to be exponentially distributed. This service is performed on a first come, first served basis.

The basic equations (derived in Ref. [11]) are as follows:

$$p_n = \frac{N!}{(N-n)!}\left(\frac{\rho^k}{k!}\right)\left(\frac{\rho}{k}\right)^{n-k} p_o \qquad \text{when } n > k \qquad (10.83)$$

$$p_n = \frac{N!}{(N-n)!}\;\frac{\rho^n}{n!}\;p_o \qquad \text{when } n \le k \qquad (10.84)$$

$$p_o = \left[\sum_{n=k+1}^{k} \frac{N!}{(N-n)!} \frac{\rho^n}{n!} + \sum_{n=0}^{k} \frac{N!}{(N-n)!} \left(\frac{\rho^k}{k!} \right) \left(\frac{\rho}{k} \right)^{n-k} \right]^{-1} \quad (10.85)$$

$$\rho = \frac{\lambda}{\mu} = \frac{\text{Mean arrival rate (failure)}}{\text{Mean service rate}} \quad (10.86)$$

where:

p_i = probability of i units awaiting service

k = number of repair channels or facilities

N = total number of systems

$$P_{OR} = \frac{N - \overline{n}}{N} \quad (10.87)$$

where:

P_{OR} = probability that a system is neither awaiting nor undergoing service.

$\overline{n}$ = average number of systems either awaiting or undergoing service at a given time and is defined by:

$$\overline{n} = \sum_{n=0}^{N} np_n \quad (10.88)$$

The specific procedure, which will be illustrated by an example, is as follows:

Step 1: Use Eq. (10.85) to solve for p_o

Step 2: Use p_o from Step 1 to help derive p_n for all values of $n \leq k$ by use of Eq. (10.84)

Step 3: Use p_o from Step 1 to help derive p_n for all values of $n > k$ by use of Eq. (10.83)

Step 4: For all values of n, from 0 through N, sum the terms np_n derived from Steps 1 through 3. This, per Eq. (10.88) gives $\overline{n}$, the average number of systems not ready

Step 5: Use Step 4 results and Eq. (10.87) to obtain the operational readiness probability, P_{OR}

Example 12: P_{OR} of Interceptor Squadron

An interceptor squadron contains fifteen planes (N = 15) and has available four flight line repair channels (k = 4). Each plane averages 50 operating hours per month out of 24 times 30, or 720 total available hours. Because of five-day, two-shift maintenance each failure requires an average of five clock hours (MTTR) to repair. The plane MTBF is 3.47 operating hours between failures. What is the operational readiness probability for this squadron?

We first compute the utilization factor ρ.

$$r = \frac{1}{\rho} \cdot \frac{\text{Operating hours per plane per month}}{\text{Total hours per month}}$$

$$= \frac{(5)(50)}{(3.47)(720)} = \frac{250}{2500} = 0.1$$

Step 1: Use Equation (10.85) to obtain p_o

$$p_o = \left[\sum_{n=k+1}^{N} \frac{N!}{(N-n)!} \frac{r^n}{n!} + \sum_{m=k}^{N} \frac{N!}{(N-n)!} \frac{r^k}{k!} \frac{r^{n-k}}{k} \right]^{-1}$$

$$p_o = \left[\sum_{m=0}^{4} \frac{15!}{(15-n)!} \frac{(0.1)^n}{n!} + \sum_{m=4}^{15} \frac{15!}{(15-n)!} \frac{(0.1)^4}{4!} \frac{(0.1)^{n-4}}{4} \right]^{-1}$$

The calculated results are shown in the following table:

n	Term (1)	Term (2)
0	1.0	- -
1	1.5	- -
2	1.05	- -
3	0.455	- -
4	0.1365	0.03412
5	- -	0.0375
6	- -	0.03753
7	- -	0.0337
8	- -	0.0127
9	- -	0.0189
10	- -	0.0113
11	- -	0.0056
12	- -	0.0022
13	- -	0.0006
14	- -	0.00013
15	- -	0.00000
Sum	4.1415	0.19428

$$p_o = (4.1415 + 0.19428)^{-1} = (4.3358)^{-1} = 0.2306$$

Step 2: Use Equation (10.84) and obtain p_n for n = 1 through 4.

$$P_n = \frac{N!}{(N - n)!} \frac{\rho^n}{n!} p_o$$

Thus,

$$p_1 = \frac{15!}{(15 - 1)!} \frac{(0.1)^1}{1!} (0.23) = 0.3450$$

$$p_2 = \frac{15!}{13!} \frac{(0.1)^2}{2!} (0.23) = 0.2415$$

$$p_3 = \frac{15!}{12!} \frac{(0.1)^3}{3!} (0.23) = 0.10465$$

$$p_4 = \frac{15!}{11!} \frac{(0.1)^4}{4!} (0.23) = 0.0313$$

Step 3: Use Equation (10.83) and obtain p_n for n = 5 through 15.

$$p_n = \frac{N!}{(N-n)!}\left(\frac{\rho^k}{k!}\right)\left(\frac{\rho}{k}\right)^{n-k} p_o$$

Thus,

$$p_5 = \frac{15!}{10!}\left[\frac{(0.1)^1}{4!}\right]\left(\frac{0.1}{4}\right)^1 (0.23) = 0.0086$$

$$p_6 = \frac{15!}{9!}\left(\frac{0.1^4}{4!}\right)\left(\frac{0.1}{4}\right)^2 (0.23) = 0.00214$$

Similarly,

$p_7 = 0.000486$

$p_8 = 0.000097$

$p_9 = 0.000017$

p_{10} through p_{15} are negligible probabilities.

Step 4: Sum the terms np_n for n = 0 through n = 15.

n	p_n	np_n
0	0.2300	0
1	0.3450	0.3450
2	0.2415	0.4830
3	0.1047	0.314100
4	0.0313	0.12500
5	0.0086	0.043000
6	0.00214	0.012850
7	0.000486	0.003400
8	0.000097	0.000776
9	0.000017	0.000153
10	---	---
11	---	---
12	---	---
13	---	---
14	---	---
15	---	---
Total		1.214779

Therefore from Equation (10.88):

$$\overline{n} = \sum_{n=0}^{N} np_n$$

$$= 1.215 \text{ planes which are not ready on the average}$$

Step 5: Using Step 4 results and Equation (5.87), we obtain P_{OR}, the operational readiness probability

$$P_{OR} = \frac{N - \overline{n}}{N} = \frac{15 - 1.215}{15} = \frac{13.785}{15} = 0.919$$

As can be seen, the calculations are rather laborious and best done by a computer. Figures 10.4-7 and 10.4-8 (from Ref. [10]) are a series of curves for N = 15 and N = 20 with k values ranging from 1 to 10 and 1 to 20, respectively. Note that 0.919 checks out the r = 0.1, k = 4 point on Figure 10.4-7.

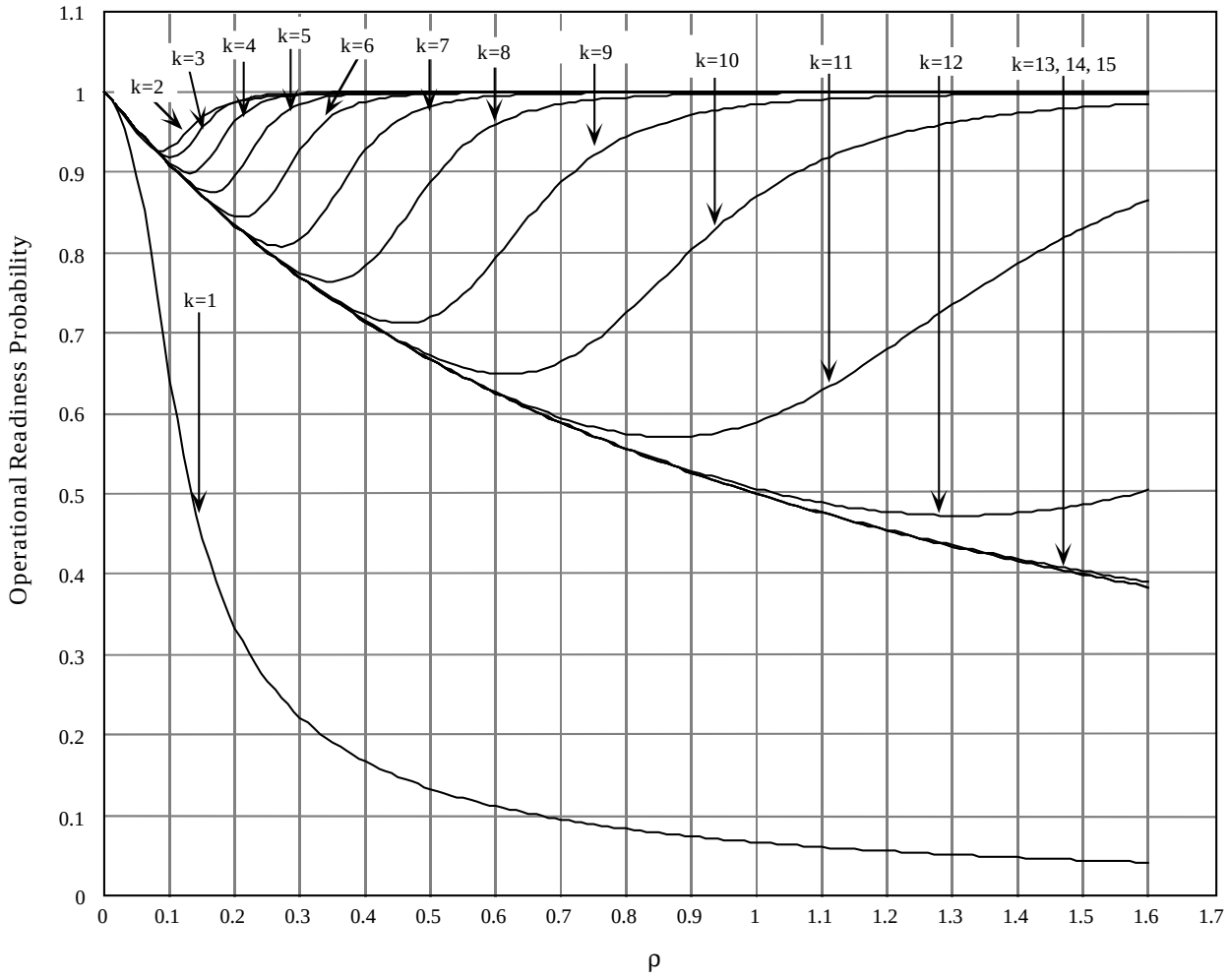


FIGURE 10.4-7: OPERATIONAL READINESS PROBABILITY VERSUS QUEUING FACTOR ρ. FOR POPULATION SIZE N = 15; NUMBER OF REPAIR CHANNELS k

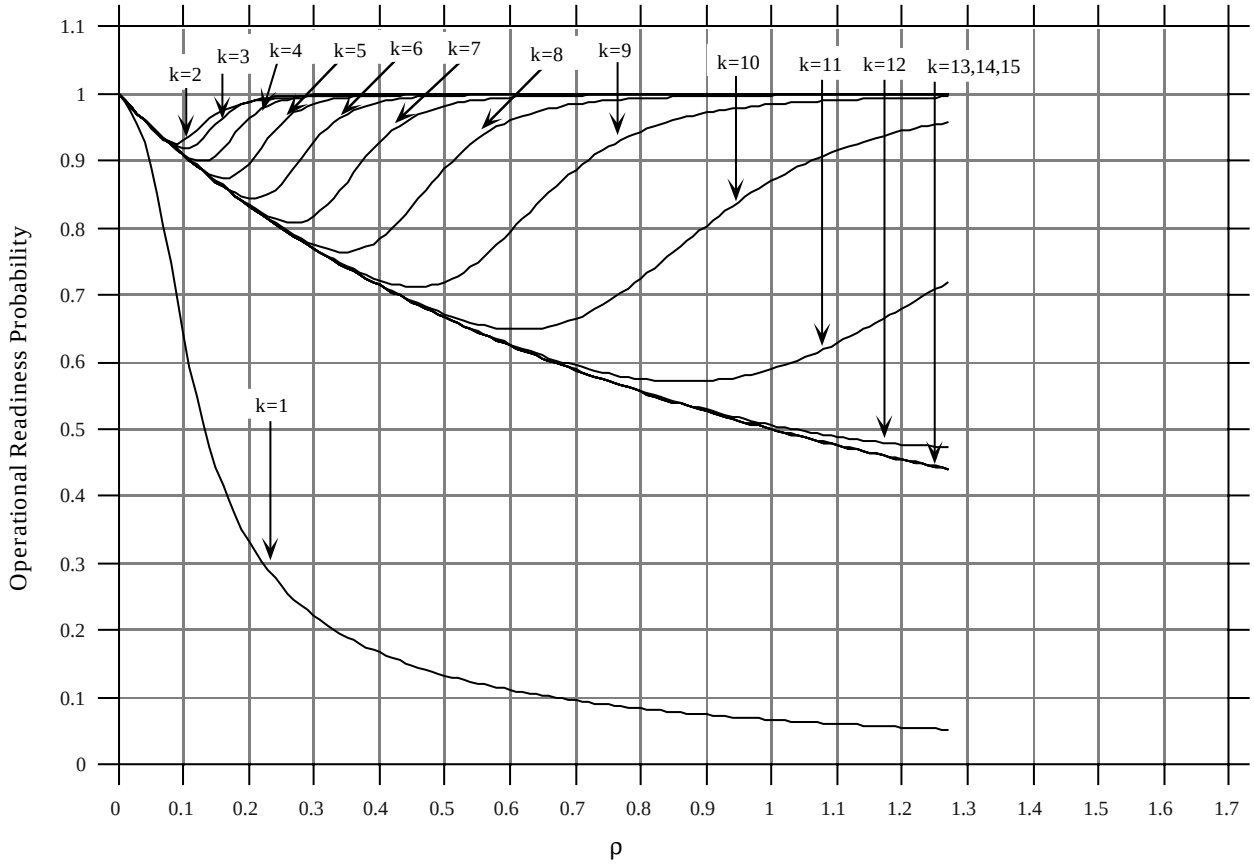


FIGURE 10.4-8: OPERATIONAL READINESS PROBABILITY VERSUS QUEUING FACTOR ρ. FOR POPULATION SIZE N = 20; NUMBER OF REPAIR CHANNELS k

10.5 Complex Models

In summing up the discussion of models, it should be recognized that there may be other measures of system R&M parameters or system effectiveness than those previously discussed. For example, in cases such as manned aircraft models it might be meaningful to combine operational readiness and equipment availability into one index, or we may wish to combine detection probability and availability for a ground radar system to be an index of the probability that a raid launched at any random time will be detected. The important point in selecting an index of system reliability effectiveness is recognizing that it is equivalent to a correct statement of the problem.

When selecting an index of effectiveness we should keep in mind some characteristics without which the index would be of little value. Probably the most important characteristic is that the index be expressed quantitatively. We should be able to reduce it to a number such that comparisons between alternative designs can be made. Furthermore, the index we choose must

have a basis in physical reality. Thus it should be descriptive of the real problem, not exaggerated or oversimplified. Yet at the same time the index should be simple enough to allow for mathematical manipulation to permit evaluating alternatives.

In complex system effectiveness mathematical models, an attempt is made to relate the impact of system reliability, maintainability, and performance to the mission profiles, scenario, use, and logistic support. Only in simple situations can a meaningful single model be developed that will relate all these parameters and yield a single quantitative measure of system effectiveness. Numerous complex computerized models exist and, as a matter of fact, every major company in the aerospace business has developed a multitude of such models.

10.6 Trade-off Techniques

10.6.1 General

A trade-off is a rational selection among alternatives in order to optimize some system parameter that is a function of two or more variables which are being compared (traded off). Examples of system trade-offs involve performance, reliability, maintainability, cost, schedule, and risk. A trade-off may be quantitative or qualitative. Insofar as possible, it is desirable that trade-offs be based on quantifiable, analytic, or empirical relationships. Where this is not possible, then semi-quantitative methods using ordinal rankings or weighting factors are often used.

The methodology for structuring and performing trade-off analyses is part of the system engineering process described in Section 4. The basic steps, summarized here are:

(1) Define the trade-off problem and establish the trade-off criteria and constraints

(2) Synthesize alternative design configurations

(3) Analyze these alternative configurations

(4) Evaluate the results of the analyses with respect to the criteria, eliminating those which violate constraint boundaries

(5) Select the alternative which best meets criteria and constraint boundaries or iterate the design alternatives, repeating Steps 2 through 5 to obtain improved solutions

System effectiveness and cost effectiveness models provide the best tools for performing trade-off studies on the system level. Through the computerized models, any changes in any of the multitude of reliability, maintainability, performance, mission profile, logistic support, and other parameters can be immediately evaluated as to their effect on the effectiveness and total cost of a system. Thus, cost effectiveness modeling and evaluation, besides being used for selecting a specific system design approach from among several competing alternatives, is a very powerful tool for performing parametric sensitivity studies and trade-offs down to component level when optimizing designs to provide the most effective system for a given budgetary and life cycle cost constraint or the least costly system for a desired effectiveness level.

At times, however, especially in the case of the more simple systems, trade-offs may be limited to achieving a required system availability while meeting the specified reliability and maintainability requirements. Comparatively simple trade-off techniques can then be used as shown in the following paragraphs.

10.6.2 Reliability - Availability - Maintainability Trade-offs

The reliability-maintainability-availability relationship provides a measure of system effectiveness within which considerable trade-off potential usually exists, e.g., between reliability, maintainability, and logistic support factors. This potential should be re-evaluated at each successive stage of system development to optimize the balance between reliability, maintainability, and other system effectiveness parameters with respect to technical risks, life cycle cost, acquisition schedule, and operating and maintenance requirements. The latter become increasingly more important as complexity of system design increases, dictating the need for integration of system monitoring and checkout provisions in the basic design.

As stated earlier in this section and in Section 2, reliability and maintainability jointly determine the inherent availability of a system. Thus, when an availability requirement is specified, there is a distinct possibility of trading-off between reliability and maintainability, since in the steady state availability depends only on the ratio or ratios of MTTR/MTBF which was previously referred to as maintenance time ratio (MTR), α , i.e.,

$$\alpha = \text{MTTR/MTBF} = \lambda/\mu \tag{10.88}$$

so that the inherent availability equation assumed the form

$$A_i = 1/(1 + \alpha) \tag{10.89}$$

As an example, consider systems I and II with

MTTR_I = 0.1 hr.

MTBF_I = 2 hr.

$MTTR_{II}$ = 10 hr.

$MTBF_{II}$ = 200 hr.

Then the steady state availability is

$$A_I = 1/\left[1 + (0.1/2) \right] = 0.952$$

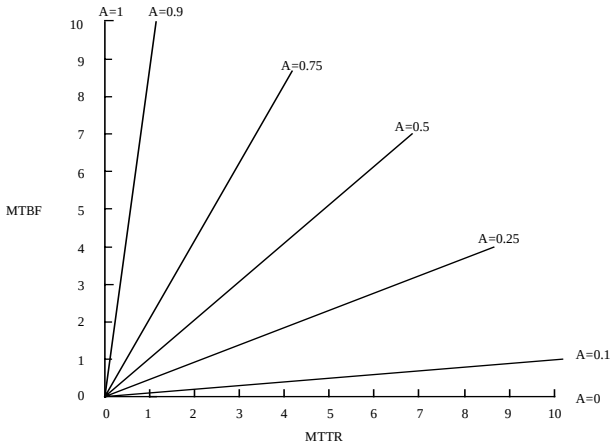
$$A_{II} = 1/\left[1 + (10/200) \right] = 0.952$$

Both systems have the same availability, but they are not equally desirable. A 10-hr MTTR might be too long for some systems, whereas a 2-hr MTBF might be too short for some systems.

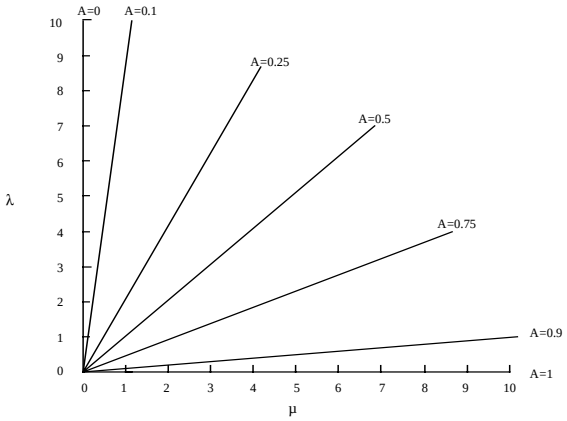
Even though reliability and maintainability individually can be increased or decreased in combinations giving the same system availability, care must be taken to insure that reliability does not fall below its specified minimum or that individually acceptable values of reliability and maintainability are not combined to produce an unacceptable level of system availability.

A generalized plot of Eq. (10.88) is given in Figure 10.6-1. A plot of A vs. λ/μ, is given in Figure 10.6-2. These equations and graphs show that in order to optimize availability it is desirable to make the ratio of MTBF/MTTR as high as possible.

Since increasing MTBF and decreasing MTTR is desirable, the equation for availability can be plotted in terms of MTBF and 1/MTTR (or μ) as shown in Figure 10.6-3. Each of the curves representing the same availability in Figure 10.6-3 just as each of the lines in Figure 10.6-1, is called isoavailability contours; corresponding values of MTBF and MTTR give the same value of A, all other things being equal. Availability nomographs useful for reliability and maintainability trade-offs are given in Reference [13]. Figure 10.6-4 is an example of an availability nomograph.



(A) AVAILABILITY AS A FUNCTION OF MTBF AND MTTR



(B) AVAILABILITY AS A FUNCTION OF λ AND μ

FIGURE 10.6-1: RELIABILITY - MAINTAINABILITY - AVAILABILITY RELATIONSHIPS

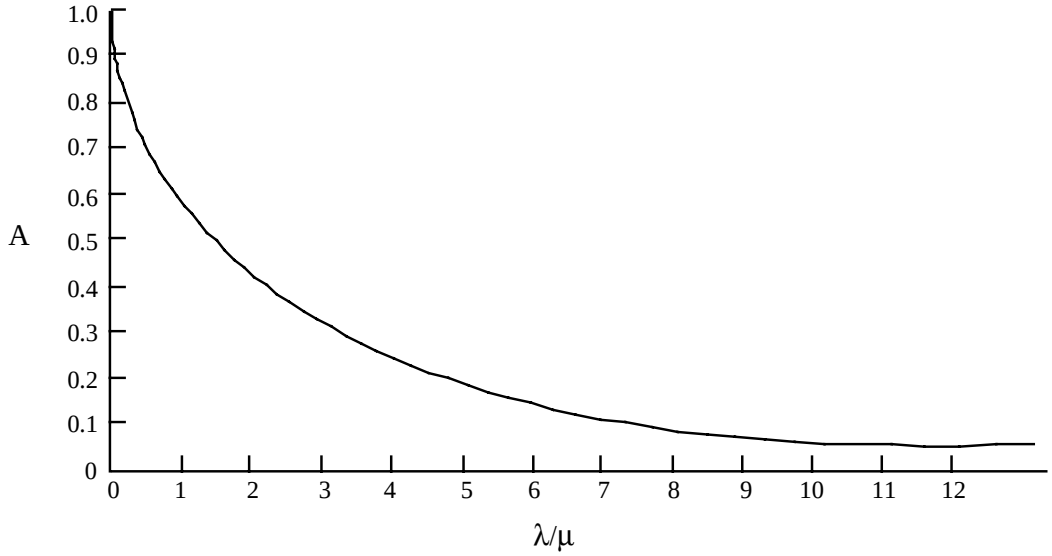


FIGURE 10.6-2: AVAILABILITY AS A FUNCTION OF λ/μ

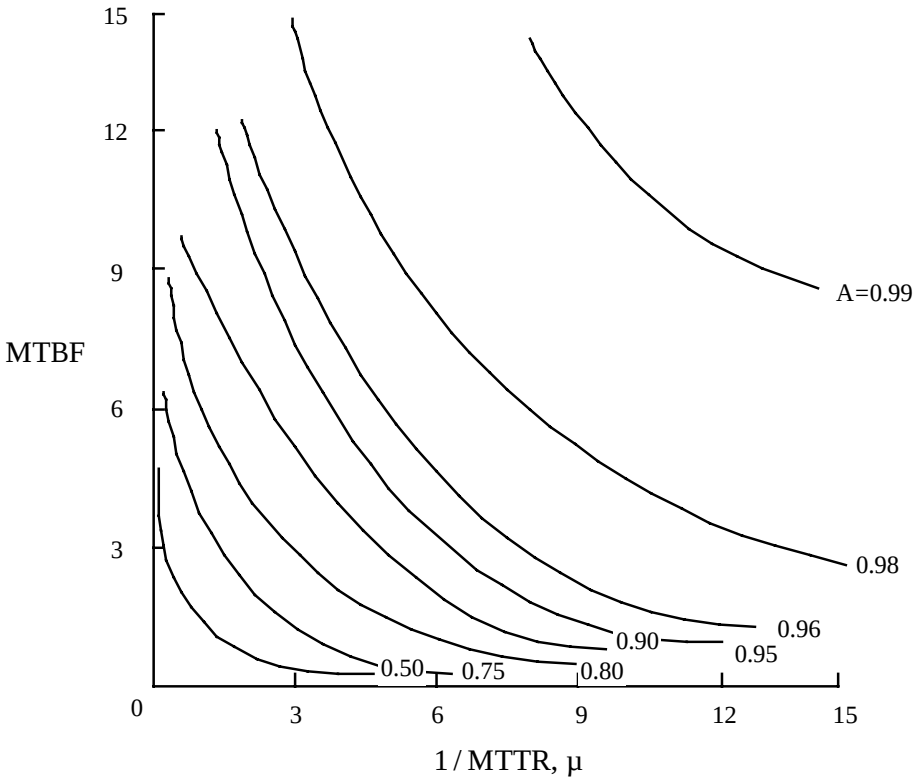


FIGURE 10.6-3: AVAILABILITY AS A FUNCTION OF MTBF AND 1/MTTR

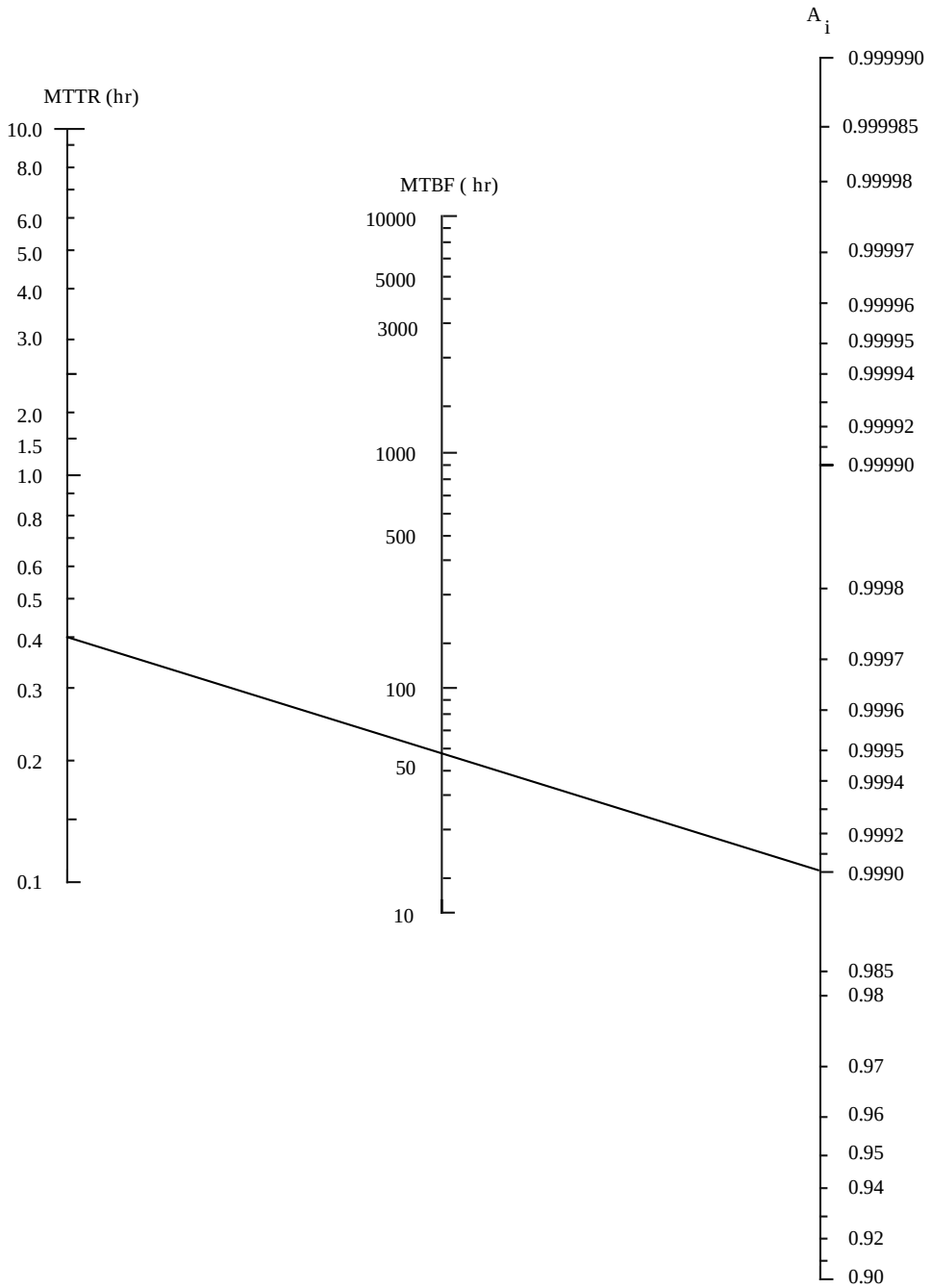


FIGURE 10.6-4: AVAILABILITY NOMOGRAPH

There are obvious practical limits which must be considered in trade-off optimization. These are called constraints, and all purposeful optimization must be bounded by constraints into feasible regions. For example, there are practical limits as to how high a value for MTBF can be achieved or how low MTTR can be made. In the one case, the reliability of system components or the required redundancy might be so high that the desired reliability could not be realistically achieved within the state-of-the-art or would be so expensive as to violate cost constraints. Similarly, MTTRs close to zero would require extreme maintainability design features, such as completely built-in test features or automatic test and checkout to allow fault isolation to each individual replaceable module, with perhaps automatic switchover from a failed item to a standby item. This also could easily violate state-of-the-art or cost constraints.

It follows, then, that trade-offs not only involve relationships among system parameters and variables but also that they are bounded by both technical and economic constraints. In a sense, all trade-offs are economic ones, requiring cost-benefit analysis (not necessarily in terms of dollar costs but rather in terms of the availability and consumption of resources, of which dollars are often the most convenient measure). Resource constraints may also include manpower and skill levels, schedule or time availability, and the technical state-of-the-art capability. Later sections of this chapter deal with the cost problem.

There are two general classes of trade-offs. In the first, the contributing system variables are traded-off against one another without increasing the value of the higher level system parameter; for example, trading-off reliability and maintainability along an isoavailability contour (no change in availability). This might be done for reasons of standardization or safety or for operational reasons such as the level at which the system and its equipments will be maintained. The other class of trade-off is one in which the system variables are varied in order to obtain the highest value of the related system parameters within cost or other constraints. For example, reliability and maintainability might be traded-off in order to achieve a higher availability. This could result in moving from one isoavailability curve to another in Figure 10.6-3, perhaps along an isocline (a line connecting equal slopes).

An example of a reliability - availability - maintainability (RAM) trade-off is given in the following paragraphs. The design problem is as follows: A requirement exists to design a radar receiver which will meet an inherent availability of 0.99, a minimum MTBF of 200 hours, and an MTTR not to exceed 4 hours. The current design is predicted to have an availability of 0.97, an MTBF of 150 hours, and an MTTR of 4.64 hours.

Using Eq. (10.88) the area within which the allowable trade-off may be made is shown by the cross-hatched portion of Figure 10.6-5. As indicated in the previous paragraph, there are two approaches which can be used for trade-off. One is to fix the availability at 0.990. This means that any combination of MTBF and MTTR between the two allowable end points on the 0.990 isoavailability line may be chosen. These lie between an MTBF of 200 hours with an MTTR of 2 hours and an MTBF of 400 hours with an MTTR of 4 hours. The other approach is to allow availability to be larger than 0.990 and thus allow any combination of MTBF and MTTR within the feasible region.

It is clearly seen that without any additional constraints the designer has a limitless number of combinations from which to choose. Assume that the following four alternative design configurations have been selected for trade-off as shown in Table 10.6-1.

Design Configuration Nos. 1, 2, and 3 all have the required availability of 0.990. Design Configuration No. 1 emphasizes the maintainability aspects in design, while Design Configuration No. 3 stresses reliability improvement. Design Configuration No. 2 is between Nos. 1 and 3 for the same availability. Design Configuration No. 4 is a combination of Nos. 1 and 2 and yields a higher availability.

Since all of these alternatives are within the feasible region shown in Figure 10.6-5 some other criterion must be used for selection of the desired configuration. In this case, we will use the least cost alternative or the one showing the greatest life cycle cost savings over the present configuration as the basis for trade-off decision. An example cost comparison of the different alternatives is shown in Table 10.6-2 (such costs would be estimated using various cost and economic models).

The cost table shows that Configuration No. 2 is the lowest cost alternative among those with equal availabilities. It also shows that Configuration No. 4, with a higher acquisition cost, has a significantly better 10-year life cycle support cost and lowest overall cost, as well as a higher availability. Thus Configuration No. 4 is the optimum trade-off, containing both improved reliability and maintainability features.

The trade-off example previously shown was a relatively simple example for the case of a single equipment. Let us now look at a more complex example. Figure 10.6-6 (a repeat of Figure 10.4-4) represents a serial system consisting of five statistically independent subsystems, each with the indicated $MTBF_i$ and $MTTR_i$. We found by the use of Eq. (10.27) that the steady state availability was:

$$A = \prod_{i=1}^{5} A_i = 0.73534$$

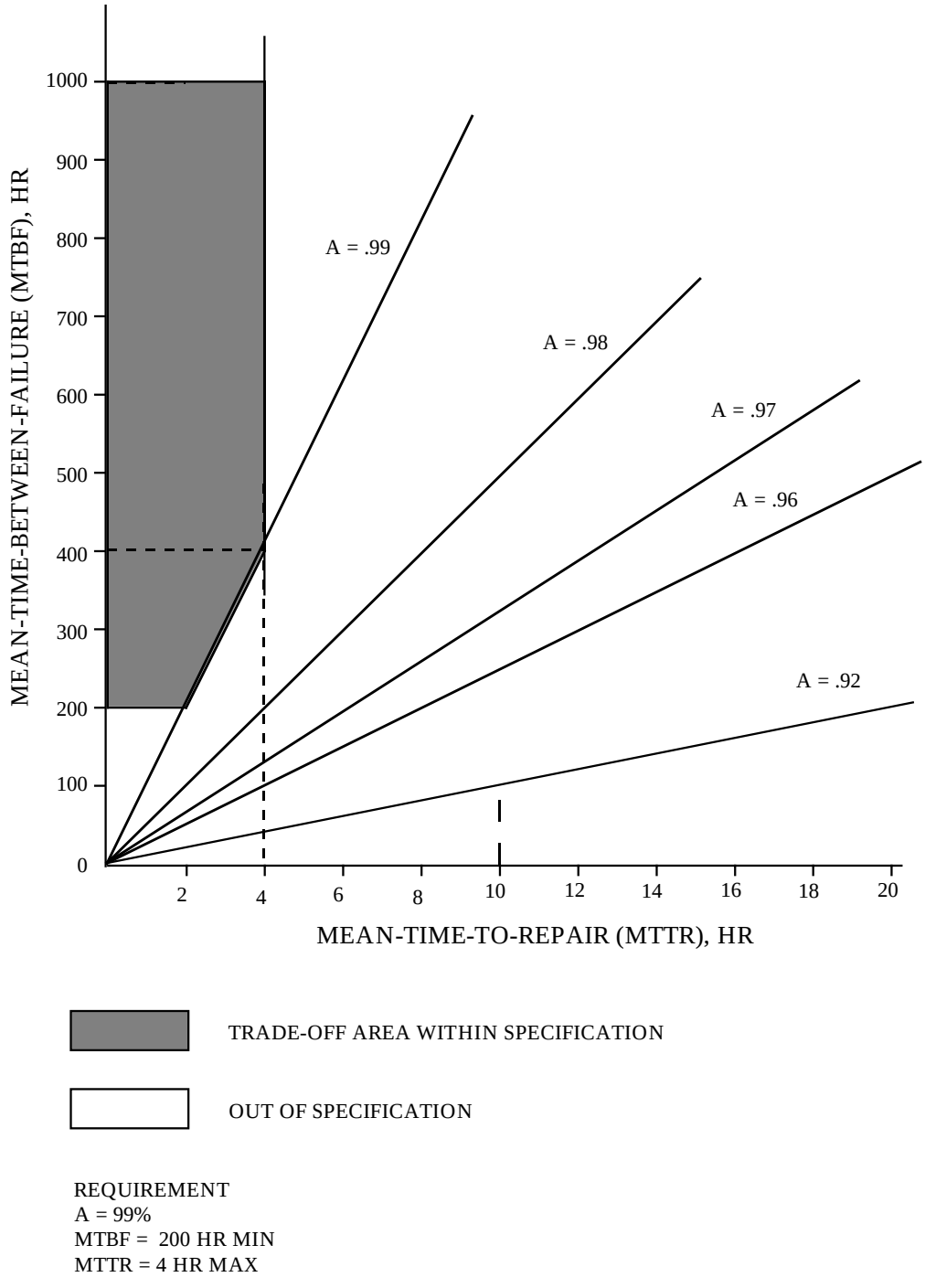


FIGURE 10.6-5: RELIABILITY-MAINTAINABILITY TRADE-OFFS

TABLE 10.6-1: ALTERNATIVE DESIGN TRADE-OFF CONFIGURATIONS

Design Configuration	A	MTBF, hr	MTTR, hr
1. R - derating of military standard parts M - modularization and automatic testing	0.990	200	2.0
2. R - design includes high reliability parts/components M - limited modularization and semi-automatic testing	0.990	300	3.0
3. R - design includes partial redundancy M - manual testing and limited modularization	0.990	350	3.5
4. R - design includes high reliability parts/components M - modularization and automatic testing	0.993	300	2.0

TABLE 10.6-2: COST COMPARISON OF ALTERNATIVE DESIGN CONFIGURATIONS

ITEM	EXISTING	1	2	3	4
Acquisition (Thousands of Dollars)					
RDT&E	300	325	319	322	330
Production	4,500	4,534	4,525	4,530	4,542
Total	4,800	4,859	4,844	4,852	4,872
10-Year Support Costs (Thousands of Dollars)					
Spares	210	151	105	90	105
Repair	1,297	346	382	405	346
Training and Manuals	20	14	16	8	14
Provisioning & Handling	475	525	503	505	503
Total	2,002	1,036	1,006	1,018	968
LIFE CYCLE COST	6,802	5,895	5,850	5,870	5,840

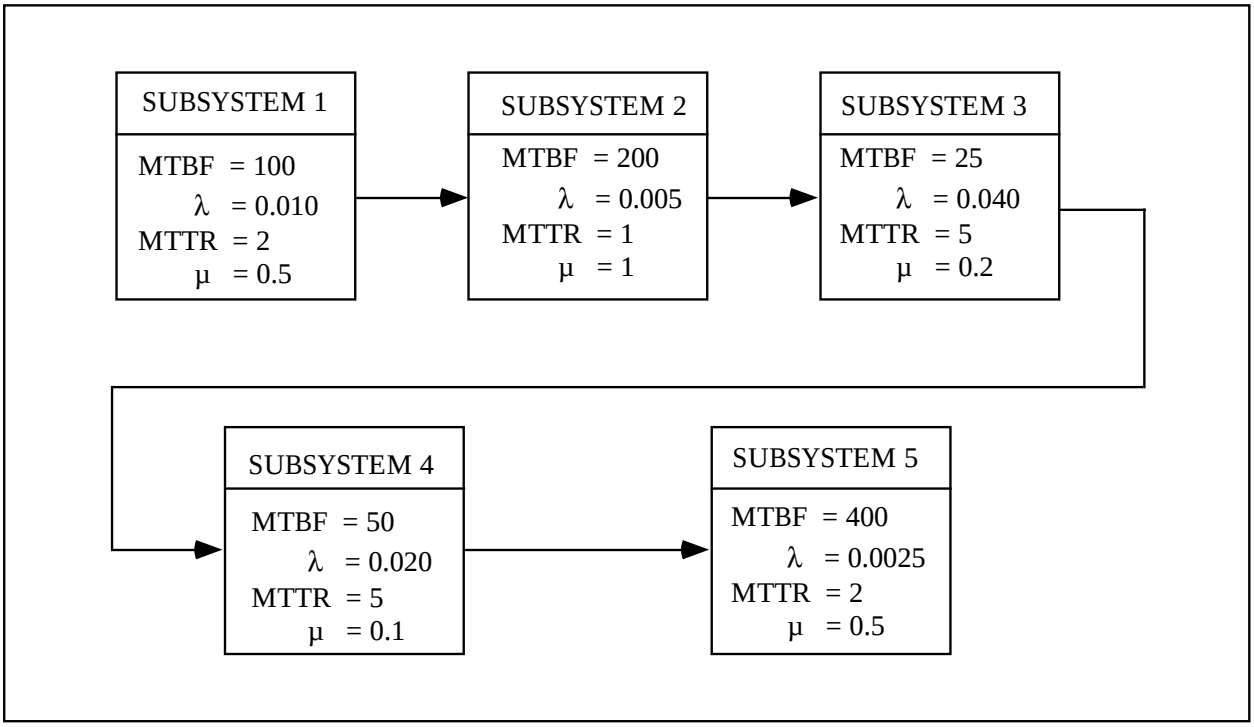


FIGURE 10.6-6: BLOCK DIAGRAM OF A SERIES SYSTEM

By inspection of the maintenance time ratios (MTRs) of each of the subsystems we note that Subsystems 3 and 4 have the lowest MTRs, given by:

$$\frac{MTTR_i}{MTBF_i} = \frac{5}{25} = 0.2$$

for Subsystem 3 and 5/50 = 0.1 for Subsystem 4. These are, therefore, the "culprits" in limiting system availability to 0.73534, which may be unacceptable for the mission at hand. If because of the state-of-the-art limitations we are unable to apply any of the design techniques detailed in Section 7 to reduce MTBF, then our first recourse is to add a parallel redundant subsystem to Subsystem 3, the weakest link in the series chain.

We shall consider two cases: (1) the case where no repair of a failed redundant unit is possible until both redundant subsystems fail and the system stops operating; or (2) repair is possible by a single repair crew while the system is operating.

For case (1) where both units must fail before repair is initiated and a single crew repairs both failed units in sequence:

$$A(1/2) = \frac{\sum_{n=1}^{2} \frac{1}{n\lambda}}{\sum_{n=1}^{2} \frac{1}{n\lambda} + \frac{n}{\mu}} = \frac{\frac{1}{\lambda} + \frac{1}{2\lambda}}{\frac{1}{\lambda} + \frac{1}{2\lambda} + \frac{2}{\mu}} \quad \text{(from Equation 10.56)}$$

$$= \frac{\frac{1}{0.04} + \frac{1}{2(0.04)}}{\frac{1}{0.04} + \frac{1}{2(0.04)} + \frac{2}{0.2}} = \frac{37.5}{47.5} = 0.789$$

This is a lower availability than the nonredundant case!

$$A_{System} = \frac{1}{1 + \frac{MTTR_{Series}}{MTBF_{Series}}} = \frac{1}{1 + .02} = 0.833 \text{ (based on Equation 10.18)}$$

For case (1), where both units must fail before repair is initiated and two repair crews simultaneously repair both failed units:

$$A(1/2) = \frac{\sum_{n=1}^{2} \frac{1}{n\lambda}}{\sum_{n=1}^{2} \frac{1}{n\lambda} + \frac{1}{\mu}} = \frac{\frac{1}{0.04} + \frac{1}{2(0.04)}}{\frac{1}{0.04} + \frac{1}{2(0.04)} + \frac{1}{0.2}} = \frac{37.5}{42.5} = 0.882$$

which is a slight improvement over the nonredundant case.

For case (2), where a single repair crew initiates repair action on a redundant subsystem as soon as it fails

$$A = \frac{\mu^2 + 2\mu\lambda}{\mu^2 + 2\mu\lambda + \lambda^2} \quad \text{(from Table 10.4-1)}$$

$$= \frac{(0.2)^2 + 2(0.2)(0.04)}{(0.2)^2 + 2(0.2)(0.04) + (0.04)^2}$$

$$= \frac{0.04 + 0.016}{0.04 + 0.016 + 0.0016} = \frac{0.056}{0.0576} = 0.972$$

as compared to 0.833 where no redundancy was used.

This corresponds to an increased system availability of

$$A = 0.73534\left(\frac{0.972}{0.833}\right) = 0.86$$

If this new value is still not acceptable redundancy might have to be applied to Subsystem 4. For example, let us use a 2-unit standby configuration for Subsystem 4 with multiple repairs; then (from Table 10.4-1), the steady state availability would be:

$$A = \frac{2\mu^2 + 2\mu\lambda}{2\mu^2 + 2\mu\lambda + \lambda^2} = \frac{2(0.2)^2 + 2(0.2)(0.02)}{2(0.2)^2 + 2(0.2)(0.02) + (0.02)^2}$$

$$= \frac{0.08 + 0.008}{0.08 + 0.008 + 0.0004} = \frac{0.088}{0.0884} = 0.995$$

Thus, the new system availability would be:

$$A = (0.86)\left(\frac{0.995}{0.909}\right) = 0.94$$

where 0.909 was the original availability of Subsystem 4.

Note, however, that to achieve these gains in availability, repair of failed redundant units must be possible while the system is operating.

Before leaving the subject of trade-offs at the system availability level, it should be pointed out that design trade-off methodologies can also be used at lower levels of the system hierarchy to increase MTBF and reduce MTTR. These are discussed in Section 7.

10.7 Allocation of Availability, Failure and Repair Rates

The previous sections discussed how availability can be calculated for various system configurations, e.g., series, parallel, etc., and how R&M can be traded off to achieve a given availability. This section discusses methods for allocating availability (and, where appropriate, failure and repair rates) among the system units to achieve a given system availability.

The reader should keep in mind that we are concerned with systems that are maintained upon failure. For the case of non-maintained systems, e.g., missiles, satellites, etc., the methods presented in Chapter 3 are appropriate for system reliability design, prediction, and allocation.

During the initial design phase of a program, detailed information is not usually available regarding the particular equipments to be employed with the system. For example, we may know that a transmitter with certain power requirements may be designed, but we do not usually know

if it is less expensive to design for a low failure rate or a high repair rate. Unless the experience of similar, previously designed transmitters can guide our decisions, estimation of the best set of alternatives is necessary. Having developed a system configuration, a range of values of equipment failure rates and repair rates that would satisfy the system availability requirement can be initially specified. The state-of-the-art limits for these equipments may not be known or the expenditures required for improvement, but we can specify their ratio, which would allow considerable freedom through the design process.

10.7.1 Availability Failure Rate and Repair Rate Allocation for Series Systems

Several cases can be considered:

(1) A single repairman must repair any one of n identical, statistically independent subsystems in series. The ratio of failure rate to repair rate is such that there is a strong possibility that a second subsystem will fail while the first one is being repaired.

(2) Same as (1) except a repairman is assigned to each subsystem and can only work on that particular subsystem.

(3) Same as (1) except some intermediate number of repairmen, r, less than the number of subsystems is assigned. Any repairman can work on any system.

(4) Repeat cases (1)-(3) with nonidentical subsystems.

10.7.1.1 Case (1)

The steady state availability in Case (1) is from Reference [25]:

$$A_S = \frac{(\mu / \lambda)^n}{n! \sum_{i=0}^{n} \frac{(\mu / \lambda)^i}{i!}} \tag{10.90}$$

where:

μ = subsystem repair rate
λ = subsystem failure rate
n = number of subsystems in series
μ/λ = "operability ratio" as opposed to λ/μ (the utilization factor)

For example, if n = 4 and A_S = 0.90, the allocation equation becomes

$$0.90 = \frac{(\mu/\lambda)^4}{24\left[1+\frac{\mu}{\lambda}+\frac{1}{2}\left(\frac{\mu}{\lambda}\right)^2+\frac{1}{6}\left(\frac{\mu}{\lambda}\right)^3+\frac{1}{24}\left(\frac{\mu}{\lambda}\right)^4\right]}$$

or $\mu/\lambda = 38.9$

The complexities of allocating failure and repair rates for even simple examples are apparent. If the subsystems are not identical, the allocation must be solved using the state matrix approach to compute availability.

10.7.1.2 Case (2)

This represents the situation in which a repairman is assigned to each subsystem. It is equivalent to the condition in which $\mu / \lambda >> 1$, i.e., failure rate is much smaller than repair rate. Since this is true of many practical systems, a wide variety of practical problems can be solved.

It was previously shown that for this case,

$$A_s = (A_i)^n = \left[\frac{1}{1 + (\lambda/\mu)}\right]^n \qquad (10.91)$$

where:

A_i = subsystem availability

n = number of subsystems

From Eq. (10.91)

$$A_i = (A_s)^{1/n} \qquad (10.92)$$

Example 13:

Three identical series subsystems must operate so as to give a total system availability of 0.9. What requirement should be specified for the availability of each subsystem? For the ratio of μ/λ for each subsystem?

$$A_i = (0.9)^{1/3} = 0.965$$

$$0.965 = \frac{1}{1 + \lambda/\mu}$$

$$\lambda/\mu = \frac{1}{0.965} - 1 = 0.036$$

Example 14:

A system consists of three identical, independent subsystems connected in series. The availability requirement is 0.99, and the repair rate is limited to 0.3 per hr. What is the minimum failure rate which must be allocated to each subsystem to satisfy the system requirement? A repairman is assigned exclusively to each subsystem.

If for Case (2) the series equipments are not identical the relationship

$$A_S = \prod_{i=1}^{n} A_i \qquad (10.93)$$

can be used to derive the individual subsystem availabilities.

Procedure	Example
(1) State the system availability requirement.	$A_s = 0.99$
(2) Compute the availability of each subsystem by $A_i = (A_S)1/n$	$A_i = (0.99)^{\frac{1}{3}}$ $= 0.99666$
(3) For each subsystem compute the ratio λ / μ by: $\frac{\lambda}{\mu} = \frac{1}{A_i} - 1$	$\lambda / \mu = \frac{1}{0.99666} - 1$ $= 0.00336$
(4) Compute λ from the previous equation with $\mu = 0.3$ per hr. The final answer is rounded off to 2 significant figures to avoid implying too much accuracy.	$\lambda = 0.00336$ x (0.3 per hr) $= 1.0$ per 1000 hr

Example 15: (using Eq. (10.93))

Three subsystems must operate to give a total system availability of 0.9. Subsystem 1 has an availability of 0.99. What should be specified for the availability of each of the other two subsystems if: (1) they are equally important, or (2) Subsystem 3 should have twice the availability of Subsystem 2 (this is interpreted as Subsystem 3 having one-half the unavailability of Subsystem 2).

(1) $A_s = 0.99\, A_2 A_3$

$A_2 = A_3$

$0.9 = 0.99(2)A_2$

$A_2 = \sqrt{0.91}$

$A_2 = A_3 = 0.954$

(2) $(1 - A_2) = 2\,(1 - A_3)$

$1 - A_2 = 2 - 2A_3$

$$A_3 = \frac{A_2 + 1}{2}$$

$$0.9 = 0.99\, A_2 A_3 = 0.99\, A_2 \left(\frac{A_2 + 1}{2} \right) = 0.99\, A_2 \left(\frac{A_2^2}{2} + \frac{A_2}{2} \right)$$

$$2 \left(\frac{0.9}{0.99} \right) = A_2^2 + A_2$$

$A_2^2 + A_2 - 1.82 = 0$

$A_2 = 0.94$

$A_3 = \frac{0.94 + 1}{2} = 0.97$

The failure and repair rate allocations for A_2 and A_3 would be

$$\lambda_2/\mu_2 = \frac{1}{A_2} - 1 = \frac{1}{0.94} - 1 = 0.064$$

$$\lambda_3/\mu_3 = \frac{1}{A_2} - 1 = \frac{1}{0.97} - 1 = 0.03$$

The previous example can be expanded to use weighting factors to derive the required subsystem availabilities. The choice of weighting factor would depend upon the system being analyzed and the significant parameters affecting availability. Some examples of weighting factors might be relative cost or equivalent complexity of the subsystem. The latter, for example, should correlate somewhat with increasing failure and repair rates. Let us examine an example of an allocation using equivalent complexity.

Example 16:

A ground surveillance series system consists of a radar, a data processor, and display subsystem. A system availability of 0.995 is required. Based upon past experience and engineering analysis, it is estimated that the complexity of each subsystem is as follows:

Display Subsystem	$\approx$	1000 component parts
Radar Subsystem	$\approx$	2500 component parts
Data Processor Subsystem	$\approx$	5500 component parts

What availability requirement should be specified for each of the subsystems to meet the system requirement?

The weight assigned to each subsystem is given by:

$$W_i = \frac{\text{Number of parts for subsystem i}}{\text{Total number of parts in system}}$$

$$W_1(\text{Display}) = \frac{1000}{1000 + 2500 + 5500} = 0.11$$

$$W_2(\text{Radar}) = \frac{2500}{1000 + 2500 + 5500} = 0.28$$

$$W_3(\text{Data Processor}) = \frac{5500}{1000 + 2500 + 5500} = 0.61$$

If the system availability requirement is 0.995, then 1 - 0.995 = 0.005 is the unavailability of the system. Using the weights previously derived to apportion the system unavailability to each of the subsystems, we get:

Display	=	(0.11)(0.005)	=	0.00055
Radar	=	(0.28)(0.005)	=	0.00140
Data Processor	=	(0.61)(0.005)	=	0.00305
SYSTEM UNAVAILABILITY			=	0.005

Thus, the required availabilities for each subsystem would be

A_1 (Display)	=	1 - 0.00055	=	0.99945
A_2 (Radar)	=	1 - 0.0014	=	0.9986
A_3 (Data Processor)	=	1 - 0.00305	=	0.99695

Verifying that the system requirement will be met

$$A_s = (0.99945)(0.9986)(0.99695) = 0.995$$

Also, as was previously shown, failure and repair rate allocation can be derived:

$$\lambda_1/\mu_1 = \frac{1}{A_1} - 1 = \frac{1}{0.99945} - 1 = 5.5 \cdot 10^{-4}$$

$$\lambda_2/\mu_2 = \frac{1}{A_2} - 1 = \frac{1}{0.9986} - 1 = 1.4 \cdot 10^{-3}$$

$$\lambda_3/\mu_3 = \frac{1}{A_3} - 1 = \frac{1}{0.99695} - 1 = 3.0 \cdot 10^{-3}$$

Another slight variation of Case (2) (Section 10.7.1.2) is a series system with nonidentical subsystems, in which each subsystem's

$$\lambda_i/\mu_i < 0.1$$

The availability of such a system with subsystems whose failures and repair are statistically independent is:

$$A_s = \frac{1}{1+\sum_{i=1}^{n} \alpha_i} \qquad (10.94)$$

where:

α_i = λ_i/μ_i with all $\alpha_i < 0.1$

n = number of subsystems in series

$$\alpha_{(system)} = \alpha_1 + \alpha_2 + ... + \alpha_n \quad (10.95)$$

To design such a system, one merely allocates the subsystem α_i's according to some weighting scheme. For example, there may be a requirement to design a new system with higher availability which is similar in design to the old system, where the relative weighting factors are the same for each new subsystem.

$$W_i = \frac{\alpha_i \text{ (new)}}{\alpha_i \text{ (old)}} \quad (10.96)$$

Example 17:

A system consisting of two statistically independent subsystems has an availability of 0.90. Subsystem 1 has an availability of 0.97, and subsystem 2 has an availability of 0.93. A new system, similar in design to this one, must meet a required 0.95 availability. What are the new subsystem availabilities and ratios of failure-to-repair rate?

Since the allocated ratios are known, additional trade-off studies can be performed to optimize λ_i and μ_i for each subsystem.

10.7.2 Failure and Repair Rate Allocations For Parallel Redundant Systems

A system comprising several stages of redundant subsystems whose λ/μ ratio is less than 0.1 can be treated as if the stages were statistically independent. The system steady-state availability A_s is:

$$A_s = A_1 \cdot A_2 \cdot A_3 \cdot ... \cdot A_n$$

where:

A_i = the availability of State I

The procedure for allocating the failure and repair rates and the availability is as follows.

Procedure	Example
(1) State the availability requirement A_s of the new system	$A_s = 0.95$
(2) Compute the sum α_s of the α = ratios for the old system $\alpha_s(\text{old}) = \alpha_1 + \alpha_2$	(Remember $\alpha_i = \lambda_i/\mu_i = \frac{1}{A_i} - 1$) $\alpha_s(\text{old}) = 0.0309 + 0.0753 = 0.01062$
(3) Compute the relative weights W_i by Eq. (10.96)	$W_1 = \frac{0.0309}{0.1062} = 0.291$ $W_2 = \frac{0.0753}{0.1062} = 0.709$
(4) Compute an overall A_s for the new system by: $\alpha_s'(\text{new}) = \frac{1}{A_s}, \quad - 1$	$\alpha_s' = \frac{1}{0.95} - 1 = 0.0526$
(5) Compute the allocated α_i' for each subsystem of the new design by: $\alpha_i' = W_i \alpha_s'$	$\alpha_1 = (0.291)(0.0526) = 0.0153$ $\alpha_2' = (0.709)(0.0526) = 0.0373$
(6) Compute the availabilities A_i' allocated to each subsystem by: $A_i' = \frac{1}{1 + \alpha_i'}$	$A_1' = \frac{1}{1 + 0.0153} = 0.985$ $A_2' = \frac{1}{1 + 0.0373} = 0.964$
(7) Check the allocated availability A_s of the new system by: $A_s' = A_1' \cdot A_2'$	$A_s = (0.985)(0.964) = 0.95$

This is equivalent to treating each stage as if it had a repairman assigned to it. It is also equivalent to saying that a single repairman is assigned to the system but that the probability of a second failure occurring while the first is being repaired is very small. If the stages are not statistically independent, the system availability must be computed by the state matrix approach. In either case, the system requirement can be obtained with a range of failure and repair rates. Trade-off procedures must be used to determine the best set of these parameters.

It will be recalled (from Eq. (10.52)) that the steady-state measure of availability for a stage where at least m out of n equipments must be available for the stage to be available can be expressed by the binomial expansion

$$A_S = \sum_{i=m}^{n} \binom{n}{i} A^i (1 - A)^{n-i} \tag{10.97}$$

and, where m = 1, i.e., only one equipment of n need be available at any one time, Eq. (10.97) simplifies to:

$$A_S = 1 - (1 - A)^{n-1} \tag{10.98}$$

If Eq. (10.97) can be expressed in terms of the operability ratio μ/λ, the initial allocation may be made. Eq. (10.97) can be expressed in terms of the operability ratio as:

$$A_S = \sum_{i=m}^{n} \frac{\binom{n}{m} (\mu / \lambda)^i}{(1 + \mu / \lambda)^n} \tag{10.99}$$

Now if a value of A_S is specified and we know the system configuration (at least how many equipments out of n-equipments must be available within each stage), we can solve for the operability ratios μ/λ.

For example, consider Table 10.7-1, in which the system availability requirement of 0.992 has been allocated to each of 4 series subsystems (stages) as indicated in column (2). In turn, in order to achieve the given stage availability, it has been determined that parallel redundant subsystems are required for each stage (column (3)) in which at least one of the redundant subsystems per stage must be available for the system availability requirement to be met.

TABLE 10.7-1: PRELIMINARY SYSTEM AND SUBSYSTEM RELIABILITY SPECIFICATIONS

(1)	(2)	(3)	(4)	(5)
Stage	Stage Availability	Number of Subsystems (n)	Number of Subsystems Required (m)	Operability Ratio
1	0.9984	4	1	4.0
2	0.9976	5	1	2.5
3	0.9984	4	1	4.0
4	0.9976	5	1	2.5

The final column (5) indicates the calculated μ/λ (operability ratio) required of each subsystem in the redundant configuration of each stage in order to achieve the allocated stage availability. Column (5) results are obtained by the use of Eqs. (10.98) or (10.99). For example, for Stage 1, m = 1, n = 4. Therefore, since m = 1, we may use Eq. (10.98).

$$A_s = 1 - (1 - A)^n$$

$$0.9984 = 1 - \left(1 - \frac{\mu}{\lambda + \mu}\right)^4$$

$$0.9984 = 1 - \left(\frac{\lambda}{\lambda + \mu}\right)^4$$

$$\frac{1}{1 + \mu/\lambda} = (1 - 0.9984)^{1/4} = 0.2$$

$$0.2\ \mu/\lambda = 1 - 0.2$$

$$\frac{\lambda}{\mu} = .25$$

This represents an upper bound of the ratio. All solutions for which the ratio ≤ .25 are acceptable.

Obviously, there are a multitude of combinations that would satisfy this equation as shown in Figure 10.7-1. Until more information becomes available concerning the cost of various failure rates and repair rates of the particular equipments involved, this initial specification allows preliminary equipment design to start with an availability goal that is consistent with the system's requirements. To facilitate calculations of operability ratio, solutions to Eq. (10.99) for n from two through five (Ref. [25]) are given in Figures 10.7-2a through 10.7-2d. The abscissa of the graphs is expressed in terms of unavailability since the plot allows for greater linearity, and, thus, ease of reading. Let us solve an example problem utilizing the graphs.

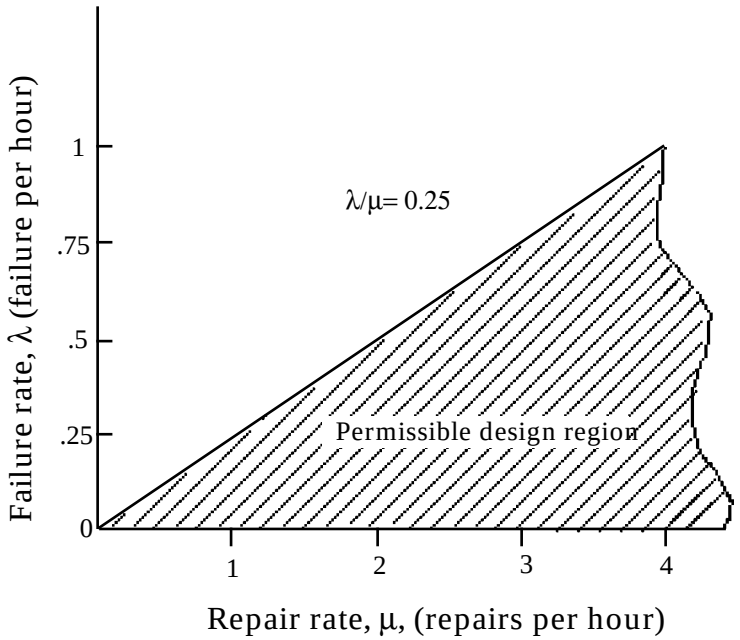


FIGURE 10.7-1: PERMISSIBLE EQUIPMENT FAILURE AND REPAIR RATES FOR $\lambda/\mu = .25$

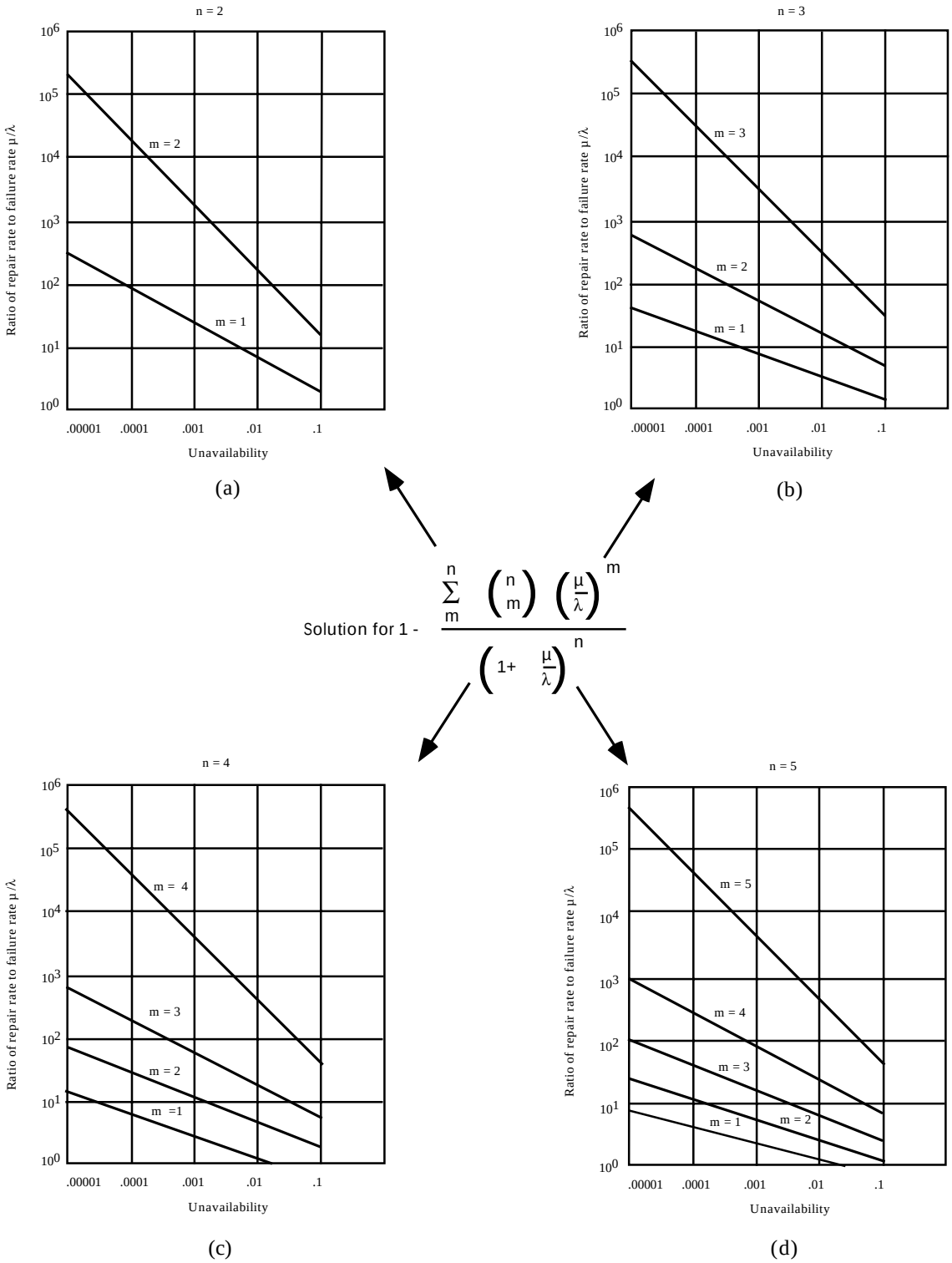


FIGURE 10.7-2: UNAVAILABILITY CURVES

Example 18:

A system consists of five identical, statistically independent subsystems connected in a parallel redundant configuration. A system availability of 0.999 is required. Four out of five subsystems must be operating for the system availability requirement to be met. What is the required λ/μ ratio? The procedure for finding this ratio is as follows.

Procedure
(1) State the system availability requirement, A_S (e.g., $A_s = 0.999$)
(2) Compute the system unavailability, U_S, by subtracting As from 1 (e.g., $U_s = 1-0.999 = 0.0010$)
(3) Enter Figure 10.7.2-2d using m = 2 and $U_S = 0.0010$, and read the required ratio (e.g., $\lambda/\mu = .01$)

10.7.3 Allocation Under State-of-the-Art Constraints

Following through the example of the previous section, we note that the allocation of an operability ratio λ/μ to each equipment does not recognize limitations on the range of either of these parameters. If R&M predictions indicate what these constraints are and they turn out to be in conflict with the preliminary allocation, revised allocations are warranted. During the reallocation, the cost of reducing the equipment failure rates and repair rates should also be considered to provide for a best balance of operability objectives. For example, in the previous section (see Table 10.7-1) the operability ratio allocated to the subsystems within the first stage was $\lambda/\mu \leq .25$. If reliability predictions indicate that a failure rate of 0.7 failures/hour can be achieved without much difficulty, this would indicate that a repair rate of at least 2.8 repairs/hour is required to meet the specifications. If, however, it is expected that repairs cannot be made at a rate greater than 2.0/hour, the specification will not be met.

As an example, let it be assumed that it is possible to design the equipment so that it can achieve a failure rate of 0.1 failures/hour - however, only at a considerable expenditure over and above that which would be required to design for a failure rate of 0.7 failures/hour. Now, it may be possible that the predicted failure rates and repair rates of the subsystems within the remaining stages are well within the operability ratio. Thus, it may be feasible to tighten the specifications of the subsystems within the other stages while relaxing the specification of the subsystems within the first stage and still achieve the required level of system availability. Again, there may be many ways of balancing the specifications. It is desirable, therefore, to choose that balance which minimizes any additional expenditure involved over that allocated for the system configuration.

Dynamic programming (Ref. [25]) is a powerful tool for balancing operability ratios in determining a system configuration at least cost.

Before leaving this subsection on allocation with redundancy, it should be pointed out that if the redundant subsystems in each stage are not identical, state matrix techniques must be used to compute availability.

10.8 System Reliability Specification, Prediction and Demonstration

Sections 6, 7 and 8 presented in great detail methods for specifying, predicting, and demonstrating system reliability.

The methods and design procedures presented in Section 7 are directly applicable to system reliability parameters for the case of non-maintained systems, e.g., missiles, satellites, "one-shot" devices, etc.

For maintained systems, the methods and procedures presented in References [26] and [50] are directly applicable to system maintainability parameters. When these are combined with the methods of Section 7 and the appropriate sections of this section, they provide a complete capability for specifying, predicting, and demonstrating most system R&M parameters, as well as trading them off to maximize system availability or some other appropriate effectiveness parameter at minimum cost.

Perhaps the only area that may need some further discussion is availability demonstration methods. At the present time no accepted test plans exist for steady state availability; however, MIL-HDBK-781 describes two availability demonstration tests; one for fixed sample size, the other a fixed time test. The tests are based upon a paper presented at the 1979 Annual Reliability and Maintainability Symposium (Ref. [26]). The paper also provides a theoretical discussion of sequential test plans, but no standardized plans are presented. Program managers or R&M engineers who wish to consider using sequential availability tests should consult the referenced paper. The proposed demonstration plans are described in the following subsection.

10.8.1 Availability Demonstration Plans

The availability tests are based on the assumption that a system can be treated as being in one (and only one) of two states, "up" or "down." At t = 0 the system is up (state X) and operates until the first failure at $T = X_1$; it is down for repairs during the restore cycle Y_1. An up/down cycle is complete by time $X_1 + Y_1$. The random variables, X_i and Y_i are each assumed to be independent and identically distributed with means E(X) and E(Y). The sequence of pairs (X_i, Y_i) forms a two dimensional renewal process.

For this system, the availability, A(t), equals the fraction of time the system is up during (0, t).

The steady state availability is

$$A = \lim_{t \to \infty} A(t) = \frac{E(X)}{E(X) + E(Y)} \tag{10.100}$$

Assume that E(X) and E(Y) and, therefore, A are unknown. Hypothesize two values of A.

$$H_o: \quad A = A_o \tag{10.101}$$
$$H_1: A = A_1 \quad \text{where } A_1 < A_o$$

On the basis of test or field data, accept or reject the hypothesis H_o by comparing the computed A to a critical value appropriate to the test type and parameters.

It is assumed that both the up and down times are gamma distributed in order to derive the relationships of each test type. However, extremely useful results can be derived assuming the exponential distribution in both cases; the exponential distribution is used in the following examples.

10.8.1.1 Fixed Sample Size Plans

This test plan is based on having the system perform a fixed number of cycles R. The result is R pairs of times-to-failure and down times (X_1, Y_1), ..., (X_R, Y_R).

Let A^R = the observed availability of the test

$$A^R = \frac{\sum_{i=1}^{R} X_i}{\sum_{i=1}^{R} X_i + \sum_{i=1}^{R} Y_i} = \frac{1}{1 + Z_R} \tag{10.102}$$

where:

$$Z_R = \frac{\sum_{i=1}^{R} Y_i}{\sum_{i=1}^{R} X_i} \tag{10.103}$$

and

A^R = the maximum likelihood estimate of A

Let,

$$\rho_o = \frac{A_o}{1 - A_o} \quad \text{under the hypothesis } H_o \tag{10.104}$$

and

$$\rho_1 = \frac{A_1}{1 - A_1} \quad \text{under the hypothesis } H_1 \tag{10.105}$$

The procedure to be followed is:

$$\text{If} \quad \rho_o Z_R > C \quad \text{reject } H_o \tag{10.106}$$

$$\rho_o Z_R \leq C \quad \text{accept } H_o$$

where C will be derived in the following paragraphs.

Assume that the up-times, X_i, are gamma distributed with parameters (m, θ) and the down times, Y_i, are gamma distributed with parameters (n, ϕ) with $n\phi = 1$.

Then it can be shown that ρZ_R is F-distributed with parameters (2nR, 2mR)

The critical value, C, and number of up/down cycles, R, are determined so that the significance test satisfies the consumer and producer risk requirements, α and β, i.e.,

$$P(\rho_o Z_R > C \,|A_o, R) \leq \alpha \tag{10.107}$$

$$P(\rho_o Z_R \leq C \,|A_1, R) \leq \beta \tag{10.108}$$

which is equivalent to:

$$C \geq F_{\alpha}(2nR, 2mR) \tag{10.109}$$

$$\frac{\rho_1}{\rho_o} C \leq F_{1-\beta}(2nR, 2mR) \tag{10.110}$$

Here $F_{\alpha}(\upsilon_1, \upsilon_2)$ denotes the upper α percentile of the F-distribution with parameters υ_1 and υ_2.

This system of inequalities has two unknowns and is solved numerically by finding the smallest integer R satisfying

$$F_{\alpha}(2nR,2mR) \bullet F_{\beta}(2mR, 2nR) \leq D$$

where D is the discrimination ratio,

$$D = \frac{A_o(1 - A_1)}{A_1(1 - A_o)} = \frac{\rho_o}{\rho_1} \tag{10.112}$$

The value of R obtained in this way is used to calculate the critical value, C:

$$C = F_{\alpha}(2nR, 2mR) \tag{10.113}$$

The OC function is

$$OC(A) = P_r\left(\rho_o Z_R \leq C \middle| A\right) = F\left(2nR, 2mR; \frac{A}{1-A} \bullet \frac{C}{\rho_o}\right) \tag{10.114}$$

where $F(\upsilon_1 , \upsilon_2 ; x)$ is the c.d.f. of the F-distribution with parameters υ_1 and υ_2.

The expected test duration is:

$$E(T) = \frac{R}{1 - A} \tag{10.115}$$

The variance of the total test duration is:

$$Var(T) = R \bullet \left\{\frac{1}{n} + \frac{1}{m} \bullet \left(\frac{A}{1 - A}\right)^2\right\} \tag{10.116}$$

For large sample size, R > 20, the distribution of T is approximately normal.

Example 19: Exponential Distribution

Let the time-to-failure and downtime distributions be exponentially distributed. Therefore, n = m = 1. Let $A_o = 0.9$ and $A_1 = 0.8$ and $\alpha = \beta = 0.2$. Calculate the parameters of the test.

Therefore,

$$\rho_o = \frac{0.9}{1 - 0.9} = 9$$

$$D = \frac{0.9(1 - 0.8)}{0.8(1 - 0.9)} = 2.25$$

Find the smallest integer R satisfying

$$F_{0.2}(2R, 2R) \leq \sqrt{2.25} = 1.5 \text{ where } F_{\alpha}(2R, 2R) = F_{\beta}(2R, 2R)$$

From a Table of Percentiles of the F-distribution we find

$$F_{0.2}(16,16) = 1.536 \text{ and } F_{0.2}(18,18) = 1.497$$

Therefore,
R = 9 satisfies the inequality

Therefore,
C = 1.497

The OC function is

$$OC(A) = F\left[18, 18;\ 0.166 \bullet \frac{A}{(1 - A)}\right]$$

10.8.1.2 Fixed-Time Sample Plans

In fixed-time sample plans, the system performs consecutive up/down cycles until a fixed-time T has elapsed. At this point, the test is terminated and the system may be either up or down. In this case the test time is fixed and the number of cycles is random.

Let A(T) = the observed availability at the end of the test.

The test procedure is

$$A(T) < A_c, \text{ then reject } H_o \qquad (10.117)$$

$$A(T) \geq A_c, \text{ then accept } H_o \qquad (10.118)$$

where the critical value A_c and test time T are chosen so that the significance test satisfies the following requirements on α and β.

$$P\left[A(T) < A_c \middle| A_o, T\right] \leq \alpha \tag{10.119}$$

$$P\left[A(T) \geq A_c \middle| A_1, T\right] \leq \beta \tag{10.120}$$

If λ_P is the upper P percentile of the standardized normal distribution and time is in mean down time units, the test time to be used is:

$$T = \left(\frac{1}{m} + \frac{1}{n}\right)\left\{\frac{\lambda_\alpha \cdot A_o\sqrt{1 - A_o} + \lambda_\beta \cdot A_1\sqrt{1 - A_1}}{A_0 - A_1}\right\}^2 \tag{10.121}$$

The critical value A_c is

$$A_c = \frac{A_o A_1\left[\lambda_\alpha\sqrt{1-A_o} + \lambda_\beta\sqrt{1-A_1}\right]}{\lambda_\alpha A_o\sqrt{1-A_o} + \lambda_\beta A_1\sqrt{1-A_1}} \tag{10.122}$$

The operating characteristic function is given by

$$OC(A) = 1 - \phi\left[\frac{A_c - A}{A \cdot \sqrt{\frac{\left(\frac{1}{m} + \frac{1}{n}\right)(1-A)}{T}}}\right] \tag{10.123}$$

where ϕ is the standardized normal c.d.f.

Example 20: Exponential Distribution

In this example use the same data as in the previous example. $A_o = 0.9$, $A_1 = 0.8$, $m = n = 1$ by the exponential assumption, $\alpha = \beta - 0.2$.

Using Eq. (10.121),

$T = 58.5$ (Mean Down Time Units)

Using Eq. (10.122),

$$A_c = 0.856$$

The OC function is

$$OC(A) = 1 - \phi \frac{0.856 - A}{A \cdot \sqrt{\frac{(1-A) \cdot 2)}{58.5}}}$$

10.9 System Design Considerations

Many of the design techniques and procedures detailed in Section 7 are directly appropriate to system design considerations.

As distinct from equipment design, system design is concerned with the broader aspects of organization and communication as they relate to the design of the individual equipment/systems. In the design of large scale systems, the need to think in terms of the whole in addition to the operation of individual equipment has become apparent. Complexity which characterizes large scale systems is at the root of the need for this broad perspective. Complex systems may perform many functions, process many inputs, translate and display many outputs, and cost a great deal of money. Therefore, only a broad perspective will permit a search for the optimum means of performing the required operations reliably.

A system R&M goal which is determined by some pertinent measure of system effectiveness stems from the system concept. Preliminary system design determines the types and minimum numbers of equipments in the network. The configuration of these equipments to achieve the system reliability goal is then determined. After a configuration is determined, an allocation of failure and repair rates is made to each equipment consistent with the system R&M goal. During the system development process, continual adjustments and re-evaluations of the means of achieving the R&M goal at least cost, are made.

The overall system design activity begins with the system concept and culminates with a set of equipment specifications that are meaningful enough to permit sound planning and comprehensive enough to present a broad perspective of the system as a single design entity. A basic philosophy of the system design is sought which allows for the determination of all important parameters in such a way that detailed design will not warrant serious redesign and the system will be optimized in its total aspect.

Equipment R&M predictions are most valuable in the early stage of a system's development. Once equipment R&M predictions are available to compare with the allocated operability ratios, discrepancies (if they exist) can be analyzed. It is also desirable to determine the expected state-

of-the-art limits of failure rate and repair rate for each equipment in the system. Thus, if predictions indicate that the operability ratio allocated to certain equipments cannot be met without additional expenditures, it may be necessary to reallocate equipment failure and repair rates such that any additional expenditures may be minimized.

Basic to the system design process is the use of comprehensive mathematical models (usually computerized) in order to optimize the system parameters to be achieved at minimum cost. There is a logical sequence to system design, an example of which is presented here for guidance:

(1) Define system R&M parameters in terms of the operational requirements of the system.

(2) Develop an index of system R&M effectiveness.

(3) Rearrange the system into convenient non-interacting stages and equipments within each stage.

(4) Apply mathematical (and statistical) techniques to evaluate alternate system configurations in terms of reliability and cost.

(5) If necessary, evaluate the consequences in terms of cost and intangible factors of each alternate configuration.

(6) Specify a system configuration, a maintenance philosophy, and the relationship with other factors (interfaces).

(7) Allocate specifications in terms of failure rate (λ) and/or repair rate (μ) to the equipment within the system as design criteria.

(8) Predict the reliability and maintainability of each equipment and the system using available data either for similar equipments or, if this is not available, from published part failure rates and estimated repair rates.

(9) Compare allocated (goal) and predicted values to determine the next best course of action.

(10) Update R&M predictions and compare with goals to allow for continuous information feedback to choose the best course of action on the system level.

The procedure is by no means rigid and should vary from system to system. However, what is important is that the systematization of objectives and the use of analytic techniques.

Since availability is a system R&M parameter which is a combined measure of reliability and maintainability, it should be maximized in the most cost effective manner. Following are some design guidelines to maximize system availability:

(1) The designed-in failure rate should be minimized, and the MTBF should be maximized.

(2) The designed-in repair rate should be maximized, and the MTTR should be minimized.

(3) Whenever possible, maintenance actions should be carried out while the equipment is running normally, thus minimizing equipment downtime.

(4) If certain functions must be shut down for maintenance, the time required for shutting down the equipment should be minimized.

(5) Should certain components require shutdowns for maintenance actions, these maintenance actions should be required as rarely as possible.

(6) Should certain maintenance actions require shutdown, the time needed for these actions should be minimized.

(7) If certain components or subsystems require shutdowns for maintenance actions, as few components as possible should be shut down.

(8) The time required for logistics should be minimized.

(9) The time required for administrative actions should be minimized.

(10) Very well written and explicitly illustrated startup and operating manuals should be prepared and made available to the users of the equipment and to the maintenance personnel.

(11) Frequent and time-consuming, prescribed, preventive maintenance actions should be minimized.

(12) Special effort should be expended to use qualified and well trained maintenance personnel; their training should be updated as required and as design changes and more modern equipment are introduced.

(13) The Reliability Design Criteria (Section 7) and the Maintainability Design Criteria given in MIL-HDBK-470.

(14) Maintenance actions which require the dismantling, moving and assembling of heavy components and equipment should be facilitated by the provisioning of special lift-off lugs and accessories.

(15) Frequently inspected, serviced, maintained, and/or replaced components should be so located in the equipment that they are more accessible and easily visible.

(16) Servicing media like lubricants, impregnates, detergents, fuels, and other consumables should preferably be supplied automatically, and waste media should be removed automatically.

(17) Whenever possible, automatic diagnostics for fault identification should be provided via failure-indicating hardware and/or special minicomputers with the associated software.

(18) There should be maximum utilization of designed and built-in automatic test and checkout equipment.

(19) The distributions of all equipment downtime categories should be determined and studied, and those maintenance actions which contribute excessively to the overall equipment downtime should be singled out and their downtimes minimized.

(20) The distributions of the equipment downtimes resulting from the failure of key components should be studied; those components contributing significantly to the overall equipment downtime should be singled out and redesigned with lower failure rates and higher repair rates.

(21) The design should be such as to achieve maximum availability at budgeted cost or acceptable availability at minimum life cycle cost.

The last item in the previous list is what it's all about - designing for maximum availability at budgeted cost or acceptable availability at minimum cost. The rest of this section is devoted to that aspect of system R&M engineering.

10.10 Cost Considerations

The most important constraint that a system designer of today must consider is cost. All of the military services face the problem of designing and fielding systems that they can "afford," i.e., which have reasonable life cycle costs (LCC). R&M have a significant impact on life cycle costs (LCC) because they determine how frequently a system fails and how rapidly it is repaired when it fails.

Thus, a vital system design consideration is how to minimize LCC by maximizing R&M within given design cost constraints.

10.10.1 Life Cycle Cost (LCC) Concepts

Life cycle cost is the total cost of acquiring and utilizing a system over its entire life span. LCC includes all costs incurred from the point at which the decision is made to acquire a system, through operational life, to eventual disposal of the system. A variety of approaches can be used

to estimate the cost elements and provide inputs to the establishment of a life cycle cost model. The total life cycle cost model is thus composed of subsets of cost models which are then exercised during trade-off studies. These cost models range from simple informal engineering/cost relationships to complex mathematical statements derived from empirical data.

Total LCC can be considered as generated from two major areas:

(1) system acquisition cost

(2) system utilization cost

In simple mathematical terms, the above can be stated by:

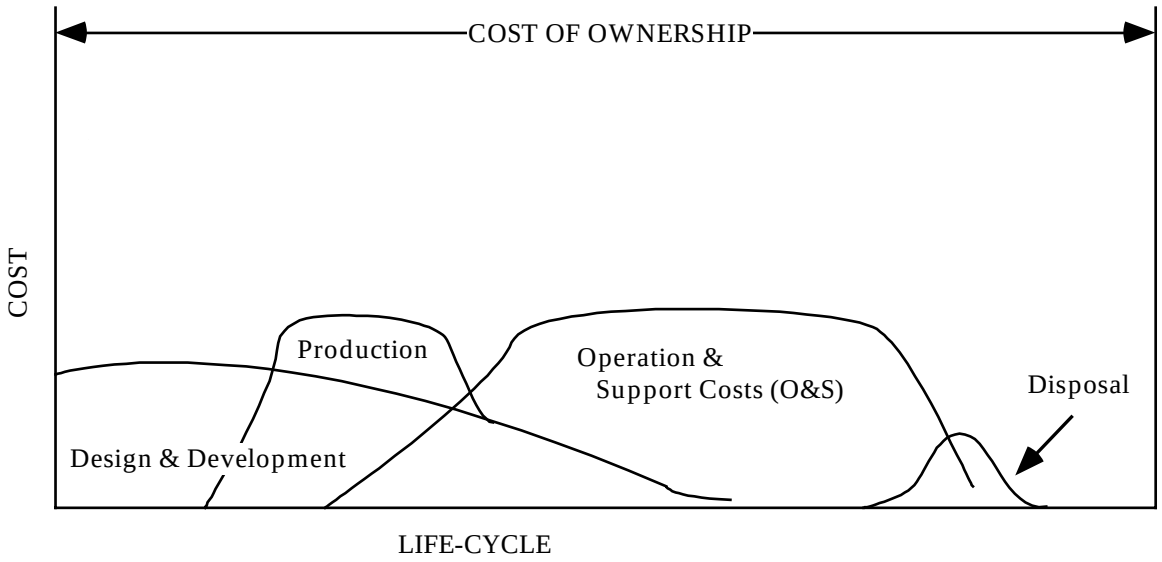
$$LCC = AC + SUC \tag{10.124}$$

where:

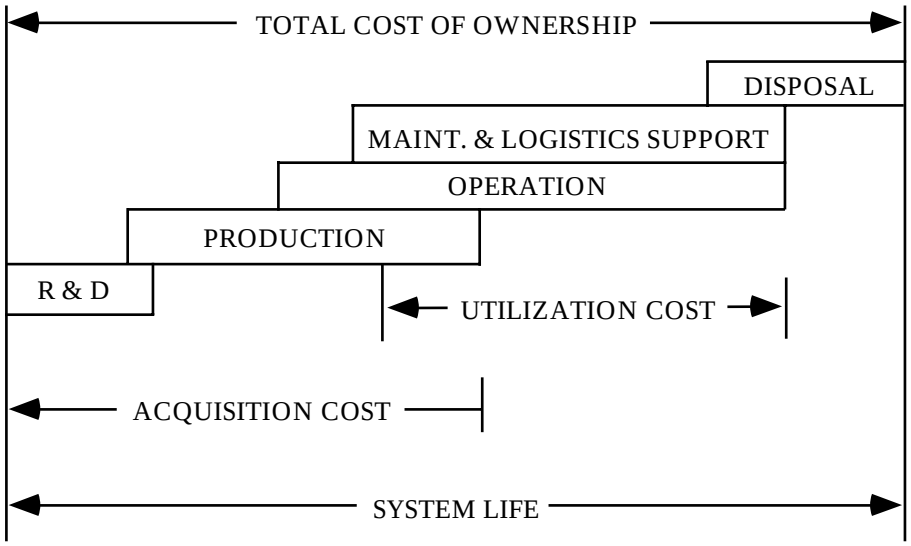
LCC = life cycle cost
AC = acquisition cost
SUC = system utilization cost

Figure 10.10-1 identifies the more significant cost categories and shows (conceptually) how LCC may be distributed in terms of the major cost categories over a system life cycle.

In general, design and development costs include basic engineering, test and system management; production costs include materials, labor, General and Administrative, overhead, profit, capitalization, handling, and transportation; operational and support (O&S) cost includes a sizable number of factors including initial pipeline spares and replacement, equipment maintenance (on/off), inventory entry and supply management, support equipment, personnel training, technical data/ documentation, and logistics management. Disposal costs include all costs associated with deactivating and preparing the system for disposal through scrap or salvage programs. Disposal cost may be adjusted by the amount of value received when the disposal process is through salvage.



(a)



(b)

FIGURE 10.10-1: LCC CATEGORIES VS. LIFE CYCLE

Life cycle cost elements are influenced by numerous system factors. Among them are:

(1) system performance requirements

(2) reliability/maintainability requirements

(3) technology

(4) system complexity

(5) procurement quantity

(6) procurement type and incentives

(7) production learning curve location

(8) maintenance and logistic support plan

Despite the emphasis on design, development and production cost in contractual requirements, the overriding objective for major DoD systems is to minimize total life cycle cost. The Government requires that life cycle costs are to be estimated during all phases of a major system acquisition program from design through operations to ensure appropriate trade-offs among investment costs, ownership costs, schedules, and performance. Trade-offs between acquisition and ownership costs as well as against technical performance and schedule must be performed in selecting from competing system design concept proposals. Life cycle cost factors are used by the Government in selecting systems for engineering and manufacturing development and production.

As shown in Figure 10.10-1, the major components of a system life cycle are its operation and support phases and the associated O&S cost. The maintenance and logistic factors that comprise O&S cost should be carefully considered and continually evaluated throughout the entire acquisition process but in particular during the conceptual phase where controllability is the greatest. These analyses are performed to provide the O&S cost impact of various design and development decisions and, in general, to guide the overall acquisition process. LCC considerations and analyses provide:

(1) A meaningful basis for evaluating alternatives regarding system acquisition and O&S cost

(2) A method for establishing development and production goals

(3) A basis for budgeting

(4) A framework for program management decisions

The application of R&M disciplines plays a key role in minimizing LCC, since one, (R), determines the frequency of failure and the other, (M), determines the time to fix a failure. System designers must balance performance, reliability, maintain- ability, and production goals in order to minimize LCC.

To meet this need, attention is focused on structuring a balanced design approach derived from a life cycle cost model that is comprised of and governed by submodels, which calculate R&M and cost variables. Figure 10.10-2 presents an overview of the R&M and cost methodology within this framework. This figure shows the life cycle cost model as the vehicle for which estimates for operation, performance, reliability, maintainability, and cost are traded off to obtain "design to" target goals which collectively represent a balanced design. This life cycle cost model includes submodels which are representative of acquisition costs and maintenance and logistics support costs, subject to the constraints of functional objectives and minimal performance requirements.

Some of the major controllable factors contributing to system life cycle cost related to these cost categories are shown in Table 10.10-1. In practice, however, all of these cost factors will not appear in each LCC analysis. Only those factors relative to the objective and life cycle phase of the analysis are included. For example, a comparison of standard commercial equipment would not include design and development costs but would include procurement and support costs. Similarly, a throwaway part or assembly would result in a simpler decision model than an item requiring on-site and off-site maintenance and repair. Thus, a system LCC decision model should be established that is flexible and capable of being exercised in various modes in keeping with the complexity of the system under analysis and the potential cost benefits to be derived from the analysis.

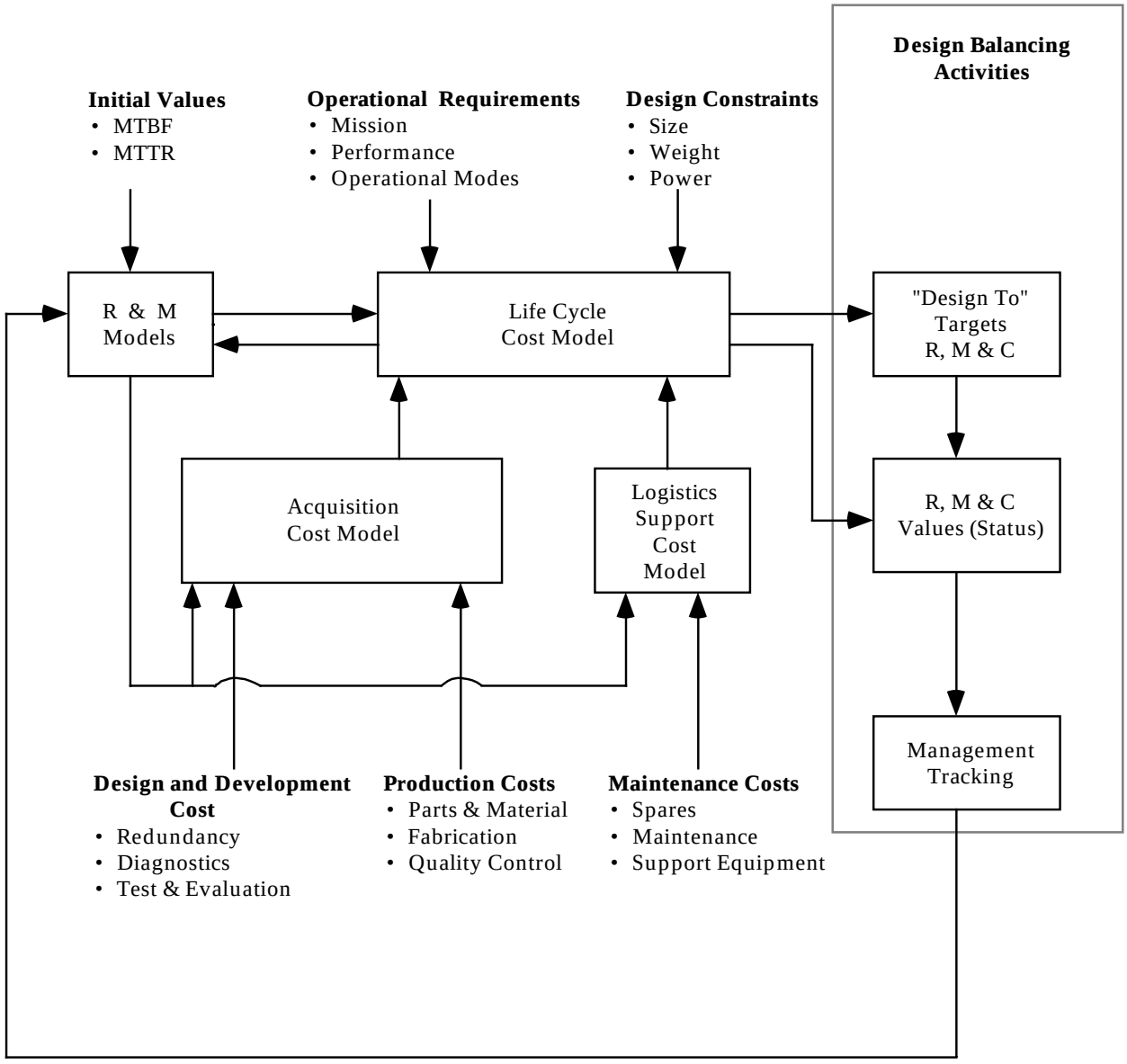


FIGURE 10.10-2: R&M AND COST METHODS

TABLE 10.10-1: LIFE CYCLE COST BREAKDOWN

Total Life Cycle Cost			
Acquisition		Operation & Support	
Basic Engineering - Design (Electrical, Mechanical) - Reliability, Maintainability - Human Factors Producibility - Component - Software Test & Evaluation - Development - $\overline{R}$ Growth - R&M Demonstration - $\overline{R}$ Screening - $\overline{R}$ Acceptance Experimental Tooling - System - $\overline{R}$ Program - $\overline{M}$ Program - Cost Manufacturing & Quality Engineering - Process Planning - Engineering Change Control - Q.A. Planning, Audits, Liaison, etc.	Recurring Production Costs - Parts & Materials - Fabrication - Assembly - Manufacturing Support - Inspection & Test - Receiving - In-process - Screening - Burn-In - Acceptance - Material Review - Scrap Rate - Rework Nonrecurring Production Costs - First Article Tests - Test Equipment - Tooling - Facilities - System Integration - Documentation (including maintenance instructions & operating manuals) - Initial spares (organizational, intermediate and depot) (pipeline)	Logistics & Maintenance Support - Pipeline Spares - Replacement Spares (organization, intermediate, depot) - On-Equipment Maintenance - Off-Equipment Maintenance - Inventory Entry & Supply Management - Support Equipment (including maintenance) - Personnel Training & Training Equipment - Technical Data & Documentation - Logistics Management - Maintenance Facilities & Power - Transportation (of failed items to and from depot)	Operation - Supply Management - Technical Data - Personnel - Operational Facilities - Power - Communications - Transportation - Materials (excluding maintenance) - General Management - Modifications - Disposal

Figure 10.10-3 illustrates (conceptually) the relationships between reliability and cost. The top curve is the total life cycle cost and is the sum of the acquisition (or investment) and O&S costs. The figure shows that as a system is made more reliable (all other factors held constant) the support cost will decrease since there are fewer failures. At the same time, acquisition cost (both development and production) is increased to attain the improved reliability. At a given point, the amount of money (investment) spent on increasing reliability will result in exactly that same amount saved in support cost. This point represents the reliability for which total cost is minimum. Consequently, reliability can be viewed as an investment during acquisition for which the return on investment (ROI) is a substantial reduction of maintenance support (the operational costs tend to remain constant regardless of reliability investment). An analogous relationship exists between maintainability and cost.

The implementation of an effective program based on proven LCC principles complete with analytical models and supporting input cost data will provide early cost visibility and control, i.e., indicate the logistics and support cost consequences of early research, development, and other subsequent acquisition decisions, such that timely adjustments can be made as the program progresses.

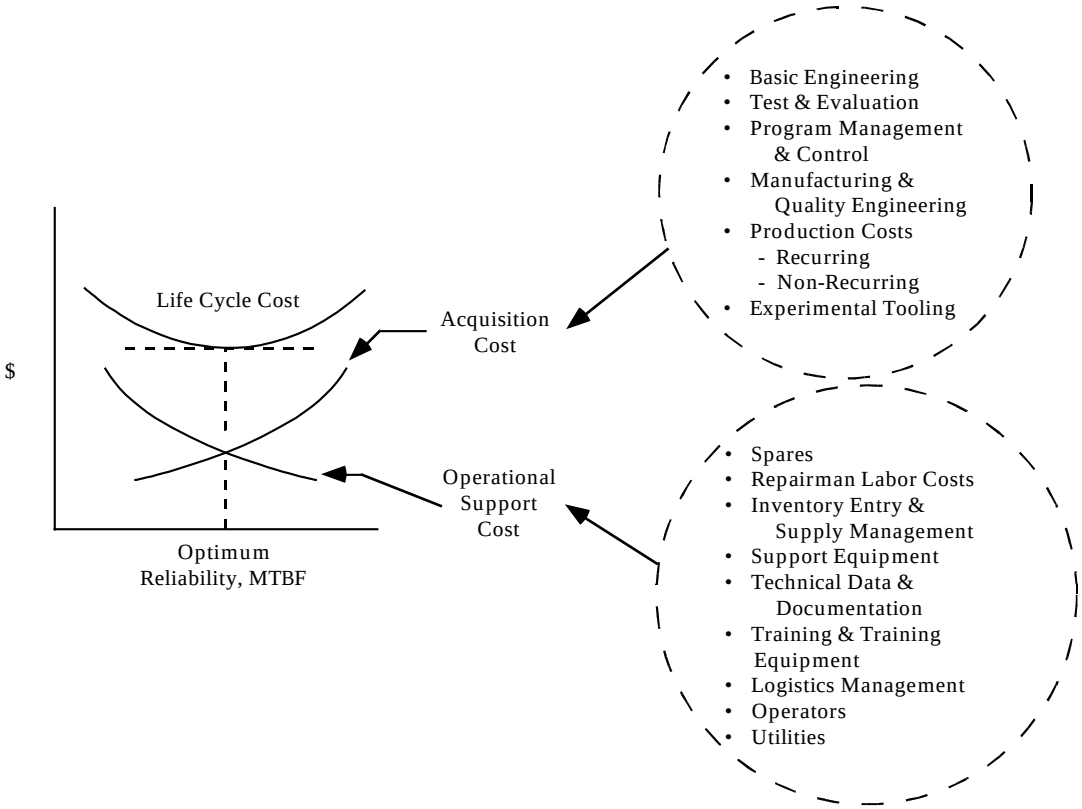


FIGURE 10.10-3: LIFE CYCLE COSTS VS. RELIABILITY

10.11 References for Section 10

1. Von Alven, W.H., Ed., Reliability Engineering. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1964.

2. AFSC-TR-65-6, Chairman's Final Report. Weapon System Effectiveness Industry Advisory Committee (WSEIAC), Air Force Systems Command, January 1965, (AD-467816), also

   AFSC TR-65-1 "Requirements Methodology," Final Report of Task Group I
   AFSC TR-65-2 "Prediction Measurement (Concepts, Task Analysis, Principles of Model Construction)," Final Report of Task Group II
   AFSC TR-65-3 "Data Collection and Management Reports," Final Report of Task Group III
   AFSC TR-65-4 "Cost Effectiveness Optimization," Final Report of Task Group IV
   AFSC TR-65-5 "Management Systems," Final Report of Task Group V

3. Elements of Reliability and Maintainability. DoD Joint Course Book, U.S. Army Management Engineering Training Agency, Rock Island, IL, 1967.

4. Systems Effectiveness. System Effectiveness Branch, Office of Naval Material, Washington, DC, 1965, (AD-659520).

5. Navy Systems Performance Effectiveness Manual. NAVMAT P3941, Headquarters Naval Material Command, Washington, DC, 1 July 1960.

7. Blanchard, B.S., "Cost Effectiveness, System Effectiveness, Integrated Logistic Support, and Maintainability," IEEE Transactions in Reliability, R-16, No. 3, December 1967.

8. Barlow, R.E., and F. Proschan, Mathematical Theory of Reliability. New York, NY: John Wiley & Sons, Inc., 1965.

9. Kozlov, B.A., and I.A. Ushakov, Reliability Handbook. Holt, Rinehart and Winston, Inc., NY, 1970.

10. Myers, R.H., K.L. Wong and H.M. Gordy, Reliability Engineering for Electronic Systems. New York, NY: John Wiley and Sons, Inc., 1964.

11. Mathematical Models for the Availability of Machine Gun Systems. Technical Report No. 3, prepared by Igor Bazovzky and Associates, Inc., for the Small Arms System Laboratory, U.S. Army Weapons Command, February 1970.

12. Availability. PAM 69-8, U.S. Army Combat Development Command Maintenance Agency, Aberdeen Proving Ground, MD, November 1970.

13. Maintainability Design Criteria Handbook for Designers of Shipboard Electronic Equipment. NAVSHIPS 94324, Department of the Navy, Change 2, 1965.

14. Orbach, S., The Generalized Effectiveness Methodology (GEM) Analysis Program. U.S. Naval Applied Science Laboratory, Brooklyn, NY, May 1968.

15. Evaluation of Computer Programs for System Performance Effectiveness, Volume II. RTI Project SU-285, Research Triangle Institute, Research Triangle Park. NC, August 1967.

16. "Computer Tells Launch Vehicle Readiness," Technology Week, April 1967.

17. Dresner, J., and K.H. Borchers, "Maintenance, Maintainability and System Requirements Engineering," Proceedings of the Third Annual Aerospace Reliability and Maintainability Conference, 1964.

18. Economos, A.M., "A Monte Carlo Simulation for Maintenance and Reliability," Proceedings of the Third Annual Aerospace Reliability and Maintainability Conference, 1964.

19. Faragher, W.E., and H.S. Watson, "Availability Analyses - A Realistic Methodology," Proceedings of the Tenth National Symposium on Reliability and Quality Control, 1964, pp. 365-378.

20. Horrigan, T.J., Development of Techniques for Prediction of System Effectiveness, RADC-TDR-63-407, Cook Electric Company, February 1964, AD-432844.

21. Maintainability Bulletin No. 8, "Maintainability Trade-Off Techniques," Electronic Industries Association, July 1966.

22. Ruhe, R.K., "Logic Simulation for System Integration and Design Assurance," Proceedings of the Third Annual Aerospace Reliability and Maintainability Conference, 1964.

23. Smith, T.C., "The Support Availability Multi-System Operations Model," Proceedings of the Third Annual Aerospace Reliability and Maintainability Conference, 1964.

24. Survey of Studies and Computer Programming Efforts for Reliability, Maintainability, and System Effectiveness. Report OEM-1, Office of the Director of Defense Research and Engineering, September 1965, AD-622676.

25. Sandler, G.H., System Reliability Engineering. Englewood Cliffs, NJ: Prentice-Hall, 1963.

26. Rise, J.L., "Compliance Test Plans for Availability," Proceedings of the 1979 Annual Reliability and Maintainability Symposium, Washington, DC, January 1979.

27. Arsenault, J.E., and J.A. Roberts, "Reliability and Maintainability of Electronic Systems," Computer Science Press, 9125 Fall River Lane, Potomac, MD 20854, 1980.

28. SD-2, "Buying Commercial Nondevelopmental Items: A Handbook," Office of the Assistant Secretary of Defense for Production and Logistics, April 1996.

29. SHARP Notes, SHARP (Standard Hardware Acquisition and Reliability Program) Program Manager, Naval Warfare Surface Center, Crane Division, Crane, IN, Code PMI.

30. "SHARP Handbook on COTS/MIL Systems," SHARP (Standard Hardware Acquisition and Reliability Program) Program Manager, Naval Warfare Surface Center, Crane Division, Crane, IN, Code PMI, 1994.

31. NAVSO P-3656, Department of the Navy Handbook for the Implementation of Nondevelopmental Acquisition.

32. MAN PRIME Handbook for Nondevelopmental Item (NDI) Acquisition.

33. ARMP-8, "Reliability and Maintainability: Procurement of Off-the-Shelf Equipment," Ministry of Defense Standard 00-40 (Part 8).

34. RADC-TR-85-91, "Impact of Nonoperating Periods on Equipment Reliability," Rome Laboratory, 1985.

35. RADC-TR-89-299, "Reliability and Maintainability Operational Parameter Translation II" Rome Laboratory, 1989.

THIS PAGE HAS BEEN LEFT BLANK INTENTIONALLY

11.0 PRODUCTION AND USE (DEPLOYMENT) R&M

11.1 Introduction

An effective system reliability engineering program begins with the recognition that the achievement of a high level of R&M in actual use is a function of design as well as all life cycle activities. Design establishes the inherent R&M potential of a system or equipment item. The transition from the computer-aided-design or paper design to actual hardware, and ultimately to operation, many times results in an actual R&M that is far below the inherent level. The degree of degradation from the inherent level is directly related to the inspectability and maintainability features designed and built into the system, as well as the effectiveness of the measures that are applied during production and storage prior to deployment to eliminate potential failures, manufacturing flaws and deterioration factors.

The impact of production, shipment, storage, operation and maintenance degradation factors on the reliability of a typical system or equipment item and the life cycle growth that can be achieved is conceptually illustrated in Figure 11.1-1. The figure depicts the development of a hardware item as it progresses through its life cycle stages. The figure shows that an upper limit of reliability is established by design, and that, as the item is released to manufacturing, its reliability will be degraded and as production progresses, with resultant process improvements and manufacturing learning factors, reliability will "grow." The figure further shows that when the item is released to the field, its reliability will again be degraded. As field operations continue and as operational personnel become more familiar with the equipment and acquire maintenance experience reliability will again "grow."

As was discussed in Section 7, reliability design efforts include: selecting, specifying and applying proven high quality, well-derated, long life parts; incorporating adequate design margins; using carefully considered, cost effective redundancy; and applying tests designed to identify potential problems. Emphasis is placed on incorporating ease of inspection and maintenance features, including use of easily replaceable and diagnosable modules (or components) with built-in test, on-line monitoring and fault isolation capabilities. During development, reliability efforts include the application of systematic and highly-disciplined engineering analyses and tests to stimulate reliability growth and to demonstrate the level of reliability that has been achieved and the establishment of an effective, formal program for accurately reporting, analyzing, and correcting failures which would otherwise occur during operation.

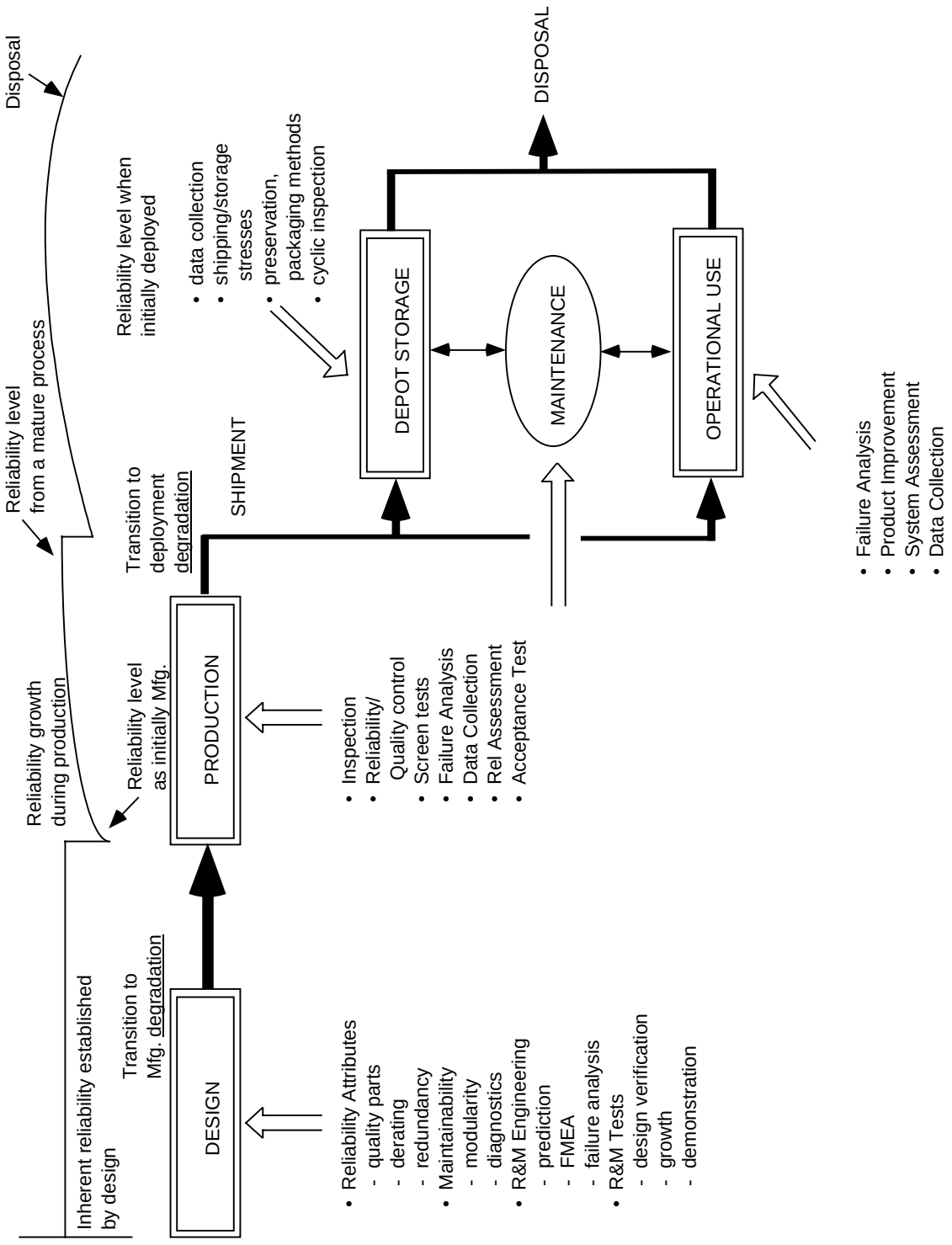


FIGURE 11.1-1: RELIABILITY LIFE CYCLE DEGRADATION & GROWTH CONTROL

Once the inherent or designed-in R&M is established, engineering efforts focus on the prevention or reduction of degradation. Well-planned and carefully-executed inspections, tests, and reliability/quality control methods are applied during production (as well as during storage and operation), to eliminate defects and minimize degradation. Manufacturing, transportation, and storage environmental stresses, as well as inspection methods and operational/maintenance procedures are continually assessed to determine the need for better inspection, screening, and control provisions to improve R&M.

This section discusses reliability degradation and growth during production and deployment. Basic procedures and guidelines are described that can be used to plan post-design reliability control measures, including the assessment and improvement of reliability during production, shipment, storage and use. Also discussed are maintainability control procedures during production and deployment.

11.2 Production Reliability Control

The need for a reliability program applicable to production becomes evident when considering that:

(1) Manufacturing operations introduce unreliability into hardware that is not ordinarily accounted for by reliability design engineering efforts.

(2) Inspection and test procedures normally interwoven into fabrication processes are imperfect and allow defects to escape which later result in field failure.

Therefore, if the reliability that is designed and developed into an equipment/system is to be achieved, efforts must also be applied during production to ensure that reliability is built into the hardware. To realistically assess and fully control reliability, the degradation factors resulting from production must be quantitatively measured and evaluated. This is particularly important for a newly fabricated item, where manufacturing learning is not yet complete and a high initial failure rate can be expected.

Since the effectiveness of inspection and quality control relates directly to reliability achievement, it would be useful to discuss basic quality engineering concepts prior to discussing specific aspects of production reliability degradation and improvement.

11.2.1 Quality Engineering (QE) and Quality Control (QC)

The quality of an item is the degree to which it satisfies the user, or it may be stated as a measure of the degree to which it conforms to specified requirements. It can be expressed in terms of a given set of attributes defined in measurable quantitative terms to meet operational requirements. Quality level can be measured by the number of defects in a given lot or item.

The purpose of a quality control program is to ensure that these attributes are defined and maintained throughout the production cycle (and continued during storage and operation). Included as part of the quality control program is the verification and implementation of inspection systems, statistical control methods, and cost control and acceptance criteria. Critical to the quality control function is the establishment of adequate acceptance criteria for individual items to assure appropriate quality protection.

Another reason for the importance of a quality control program has to do with continuous quality improvement. Measurement systems must be in place to be able to separate special problems from those that can be attributed to common causes such as random variation in the design, development and manufacturing process. Further, as stated in reference 16, “the collection and analysis of data is the key to identifying specific improvement targets. Data is the raw material that we turn into improvement projects. Data must be the basis for making every decision about improvement.” While all data are not necessarily a result of the QC program, having such a program is a key element to ensuring that data are produced and collected that can be used to ensure quality protection and provide a baseline for quality improvement.

Quality, as with reliability, is a controllable attribute which can be planned during development, measured during production, and sustained during storage and field repair actions. The achievement of acceptable quality for a given item involves numerous engineering and control activities. Figure 11.2-1 depicts some of these activities as they apply to a system over time. These activities represent an approach to a comprehensive and well rounded Quality Control Program.

Keys to ensuring the basic quality of a hardware item as depicted in Figure 11.2-1 are: the specification of cost effective quality provisions and inspections covering the acquisition of new hardware items; the storage of hardware and material; and the repair, reconditioning or overhaul of deployed items. This means that quality requirements should be included in procurement specifications, in-storage inspection requirements, and in-maintenance work requirements, as applicable, and that responsive quality programs are to be planned and implemented to meet these requirements. This section discusses quality control during the acquisition of new systems and hardware items.

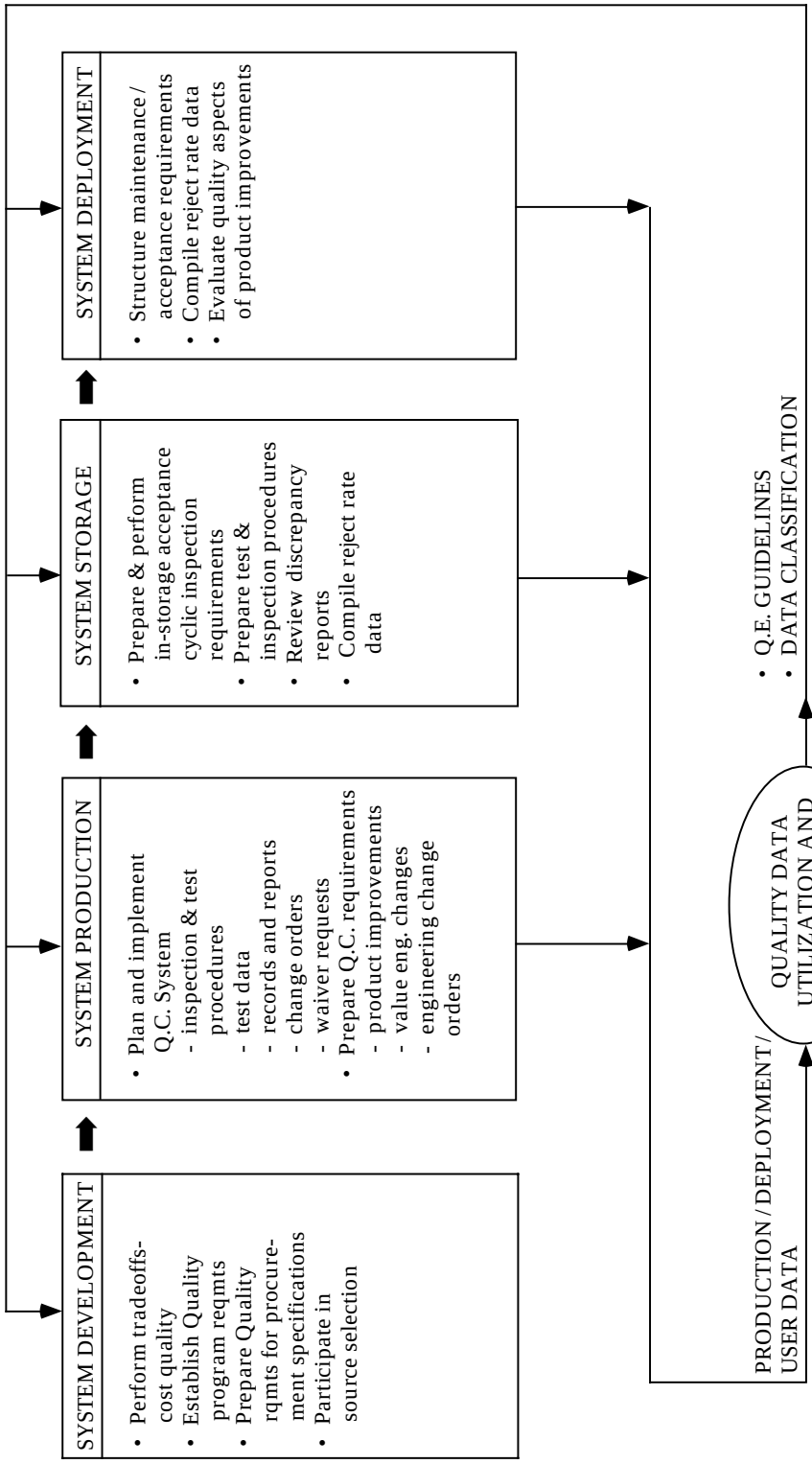


FIGURE 11.2-1: QUALITY ENGINEERING AND CONTROL OVER TIME

11.2.1.1 Quality System Requirements

Until recently, due to acquisition reform (AR) changes, quality requirements applied during acquisition generally followed Military Specification MIL-Q-9858, Quality Program Requirements. MIL-Q-9858 was the basic standard for planning quality programs for DoD development and production contracts. Under AR, MIL-Q-9858A was cancelled by Notice 2 dated October 1996. As with other canceled military documents, there is no barrier to a system developer, or contractor, using MIL-Q-9858A as a basis for a quality program, or quality system.

Prior to its cancellation, MIL-Q-9858A, Amendment 3, dated 5 September 1995 stated that for new designs, the International Organization for Standardization (ISO) 9001, ISO 9002 quality system standards, the ANSI/ASQC Q9001, ANSI/ASQC Q9002 quality system standards, or a comparable higher-level non-government quality system should be used. The ANSI/ASQC Q9000 series documents are the technical equivalent to the ISO 9000 series documents.

11.2.1.1.1 ISO 9000

Because the DoD has adopted the ANSI/ASQC Q9000 Standards Series (technical equivalent to the ISO 9000 Standards Series), it is prudent to provide some information on the ISO 9000 quality system. Adoption by the DoD means that the ANSI/ASQC Q9000 documents are listed in the DoD Index of Specifications and Standards (DODISS) and are available to DoD personnel through the DODISS publications distribution center. Note, however, that the use of the Q9000 standards has not been included within the Federal Acquisition Regulation (FAR) or the DoD FAR Supplement (DFARS). In fact, DFARS paragraph 246.102(4) states that departments and agencies shall: "Provide contractors the maximum flexibility in establishing efficient and effective quality programs to meet contractual requirements. Contractor quality programs may be modeled on military, commercial, national, or international quality standards." The last sentence allows application of MIL-Q-9858A, ISO 9000 or ANSI/ASQC Q9000 standards for quality planning.

As previously noted, ISO 9000 is a family of standards on quality. These standards have been developed by ISO Technical Committee TC 176. The family of standards is shown in Figure 11.2-2.

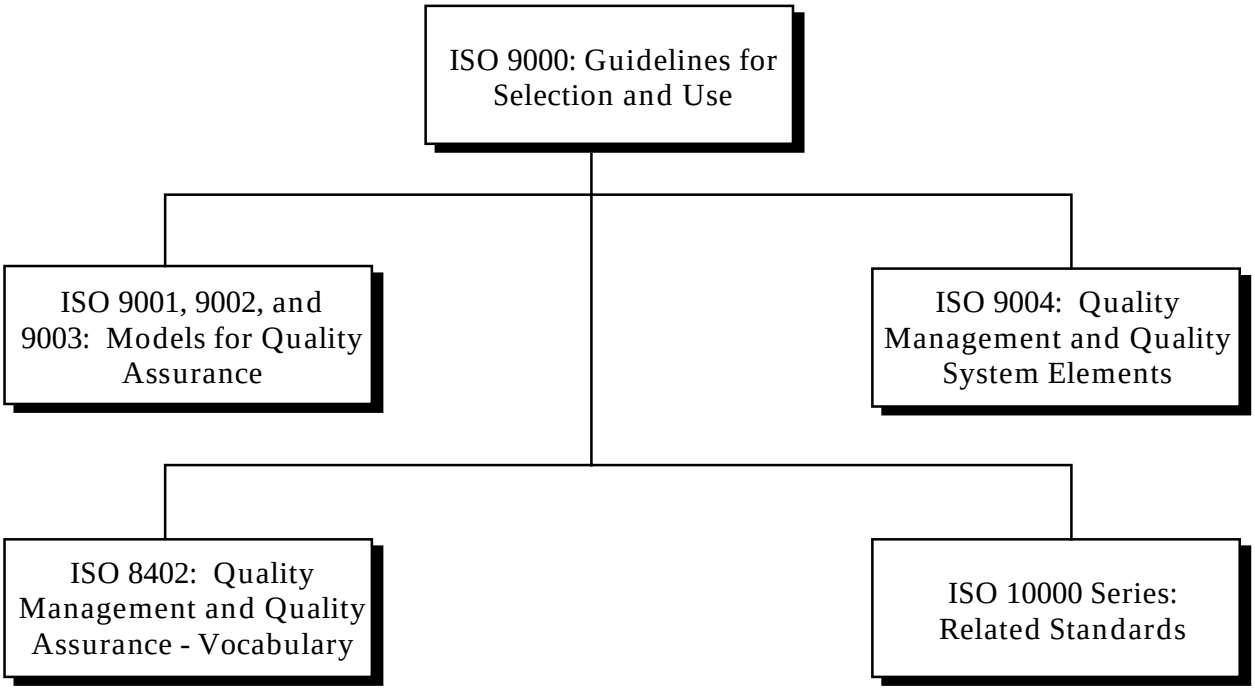


FIGURE 11.2-2: ISO 9000 FAMILY OF STANDARDS

The titles of the five standards are:

- ISO 9001: Quality Systems - Model for quality assurance in design, development, production, installation, and servicing certification system

- ISO 9002: Quality Systems - Model for quality assurance in production, installation and servicing

- ISO 9003: Quality Systems - Model for quality assurance in final inspection and test.

- ISO 9000
  - Part 1 (9000-1): Guidelines for Selection and Use
  - Part 2 (9000-2): Quality Management and Quality Assurance Standards
    - Generic guidelines for the application of ISO 9001, ISO 9002 and ISO 9003
  - Part 3 (9000-3): Quality Management and Quality Assurance Standards
    - Guidelines for the application of ISO 9001 to the development, supply and maintenance of software
  - Part 4 (9000-4): Quality Management and Quality Systems Elements - Guidelines

In reviewing the family of ISO 9000 standards, it can be seen that rather than having one, all-encompassing quality standard and relying on the user to tailor it accordingly, some tailoring has already been done. Further, there are several guidance documents (i.e., ISO 9000-1 through 9000-4) are available to assist in selecting a particular quality system standard.

11.2.1.1.1.1 Comparing ISO 9000 to MIL-Q-9858

The major thrusts behind both MIL-Q-9858 and the ISO 9000 Series Standards are essentially the same. Table 11.2-1, previously printed in MIL-HDBK-338, shows that MIL-Q-9858 covered 17 quality program elements.

As a comparison, ISO 9001 (ANSI/ASQC Q9001-1994) defines the following 20 elements of a quality system:

1. Management responsibility
2. Quality system
3. Contract review
4. Design control
5. Document and data control
6. Purchasing
7. Control of customer - supplied product
8. Product identification and traceability
9. Process control
10. Inspection and testing
11. Control of inspection, measuring and test equipment
12. Inspection and test status
13. Control of nonconforming product
14. Corrective and preventive action
15. Handling, storage, packaging, preservation and delivery
16. Control of quality records
17. Internal quality audits
18. Training
19. Servicing
20. Statistical techniques

Many of the subparagraphs within the above 20 elements cover the same subject area as did MIL-Q-9858. The MIL-Q-9858 elements are listed in Table 11.2-1.

Whereas MIL-Q-9858 recommended the use of MIL-I-45208, Inspection System Requirements, when requirements were less stringent, the ISO 9000 family of standards include ISO 9002 and ISO 9003 for use in such cases.

TABLE 11.2-1: MIL-Q-9858 QUALITY PROGRAM ELEMENTS

- *Quality Program Management*
  - Organization
  - Initial Quality Planning
  - Work Instructions
  - Records
  - Corrective Action
- *Facilities and Standards*
  - Drawings, Documentation and Changes
  - Measuring and Testing Equipment
  - Production Tooling Used as Media of Inspection
  - Use of Contractor's Inspection Equipment
  - Advanced Metrology Requirements
- *Control of Purchases*
  - Responsibility
  - Purchasing Data
- *Manufacturing Control*
  - Materials and Material Control
  - Production Processing and Fabrication
  - Completed Item Inspection and Testing
  - Handling, Storage and Delivery
- *Statistical Quality Control and Analysis*
  - Indication of Inspection Status

11.2.1.1.1.2 Why ISO 9000?

There are varied reasons for recent interest in ISO 9000, and in becoming what is called "ISO 9000 Registered," both within the US and world-wide. A detailing of the reasons and history of ISO 9000 can be found in references 17 - 19. However, a brief explanation is provided here. The development for a worldwide set of quality standards grew as each country's economy became a global one, rather than local. To meet this need, and to develop a set of standards that would be acceptable to a large number of countries worldwide, ISO, having a global membership, created the ISO 9000 series standards in 1987. The US member of ISO is ANSI.

Recently, the European Community (EC), made up primarily of the Western European powers, adopted a policy of buying "regulated products" (e.g., environmental, health, safety related) from companies that were proven to be compliant with the quality system requirements of ISO 9000.

The concept of registration has quickly spread worldwide, including to the US. Reasons for ISO implementation include pressure from customers, marketing pressures, as a vehicle for company improvement, or simply to become a qualified vendor of regulated products in the EC countries.

To become ISO 9000 registered, a company must create a quality system based on ISO 9001, 9002 or 9003, which may be a modification of an existing system. Once this is accomplished, a qualified member of a national Registrar Accreditation Board (RAB) must perform a quality audit of a candidate company's quality system to verify that it is compliant with the chosen ISO 9000 standard. See reference 17 for further information on ISO 9000 implementation.

Some final comments regarding ISO 9000 registration have to do with cost. Reference 18 notes that the cost to implement ISO 9000 for a small company can range from $12,500 to $50,000 and for a large company from $300,000 to $750,000. Reference 17 states that as of 1995, the minimum charges for a registrar was between $1,500 and $2,500 per person, per day, when working on-site. Of course, much depends on the size of the company, facilities, number of distinct product lines, and whether or not a quality system is already being used that is similar to ISO 9000. The time to implement ISO 9000 and become registered is approximately one year. Of course, ISO 9000 can be used much the same way as MIL-Q-9858 was, without going through the process of becoming registered. Note, however, that the customer will still have the right to determine if your company is compliant with the chosen quality system standard, be it ISO 9000 or any other standard.

11.2.1.2 Quality Control

A quality control program is a necessary part of most quality systems. Quality control is the operational techniques and activities that are used to fulfill the requirements for quality.

The degree of quality control for a given item is determined by considering the benefits derived from and the cost of the provisions. There are numerous examples of the considerations which apply to quality control in the production environment. These include:

(1) Sampling vs. 100% inspection

(2) Extent of quality controls during design and manufacturing

(3) Defect analysis and rework program

(4) Inspection level and expertise

(5) Special test and inspection equipment, fixtures, gauges, etc., vs. general purpose equipment

(6) Prototype tests and inspection vs. full production control

(7) Quality of purchased material

(8) Extent of quality control audits and vendor surveillance

(9) Extent of line certification

One of the basic functions of a manufacturer's quality control organization is to make tradeoff decisions relative to these considerations and to ensure that quality is adequately planned and controlled, consistent with specified requirements and the constraints of the particular system.

Accomplishment of the quality control function, like reliability, requires a comprehensive and highly detailed program comprising of effective, systematic, and timely management activities, engineering tasks, and controlled tests. The production and acceptance of high quality hardware items requires definition and implementation of an effective quality management and control program that includes:

(1) Performance of detailed quality analysis, planning and cost tradeoff analyses during hardware development.

(2) Application of systematic and highly disciplined quality control tasks during production whose purpose is to identify and correct problems during production prior to an item's release to storage and field use.

(3) The establishment of a storage/field quality and reliability assurance program. This program provides controls and procedures which allow a smooth transition from production to storage and field use without degrading the reliability/quality level. It also emphasizes nondestructive testing at critical stages in the production/storage/depot maintenance process.

Once the quality program has been planned, efforts then focus on the performance of engineering tasks on an ongoing basis to control the quality of the hardware items. Many of the manufacturer's quality engineering and control tasks are outlined in Table 11.2-2.

TABLE 11.2-2: QUALITY ENGINEERING TASKS

- Review engineering drawings and specifications, prototype test data, and R&M engineering data to determine impact on item quality.
- Review purchased material from a quality standpoint. This would include:
  - Evaluation of purchase requisitions and orders
  - Selection and certification of vendors
  - Approval of vendor component part/assembly samples
  - Review of part/material specifications (in particular, critical component identification and control)
  - Evaluation of purchased material through inspection planning, incoming inspection, and complete test data documentation control
  - Disposition and allocation of inspected material, discrepant material, review board provisions
- Evaluate material item manufacturing through a review of process inspection planning, workmanship and acceptance standards, instructions and procedures, production and QA inspection and testing.
- Determine adequacy (accuracy, calibration program, etc.) of inspection tests, production equipment, and instrumentation.
- Provide engineering direction and guidance for the acceptance inspection and test equipment in support of new item procurement production, reconditioning, and maintenance activities.
- Exercise control over the acquisition, maintenance, modification, rehabilitation, and stock level requirements of final acceptance inspection and test equipment.
- Provide product assurance representation to Configuration Control Boards, and serve as the control point for evaluation and initiation of all configuration change proposals.
- Advise, survey, and provide staff guidance for special materials and processes technology, as applied to quality control systems.
- Evaluate the adequacy, effect, and overall quality of process techniques, particularly those processes which historically have a significant impact on an item's quality.
- Evaluate reliability/quality data stemming from production, storage and use to:
  - Identify critical items having high failure rates, poor quality or requiring excessive maintenance
  - Identify significant failure modes, mechanisms, and causes of failure
  - Reduce and classify data and, in particular, define and classify quality defect codes
  - Formulate Q.C. guidelines to support preparation of procurement specifications
  - Prepare failure data and statistical summary reports
- Identify critical material items where cost effective reliability and quality improvement can be effectively implemented. Candidates for improvement include those items which have a history of poor quality, frequent failure, require extensive maintenance effort, and have excessive support costs.

TABLE 11.2-2: QUALITY ENGINEERING TASKS (CONT'D)

- Make general reliability/quality improvement recommendations on selected equipment.
- Provide product assurance engineering impact evaluations for configuration change, product improvement, and value engineering or cost improvement proposals.
- Determine the effectiveness of improvements on item reliability/quality.
- Develop calibration procedures and instructions, maintain and recommend changes to publications, equipment improvement recommendations and new calibration requirements, addressing calibration parameters.

An integral part of an effective quality control program is to make available to its engineers documented instructions, procedures, or guidance which fully describe the functions and tasks required to achieve its objective. Data collected during early production and testing activities, as well as historical data on similar systems from depot storage, maintenance actions, and field operations, can be compiled, reduced and applied to improve the production quality engineering and control activities. This data, for example, can be used to:

(1) Track quality

(2) Compare the benefits of various quality programs and activities:
- Production quality control techniques
- Vendor control and audits
- 100% inspection
- Sampling inspection
- Special quality assurance procedures

(3) Determine the effectiveness of quality control programs related to:
- Materials and materials control
- Inspection and testing of piece parts and subassemblies
- Production processing fabrication
- Completed item inspection and testing
- Handling, storage and delivery
- Corrective action implementation

(4) Determine the effects of depot storage, operation and maintenance factors:
- Depot level inspections
- Personnel
- Logistics
- Operational environment
- Mission profile

- Maintenance organization
- Quality classification codes
- Quality guidelines to support preparation of procurement specifications, storage inspection requirements and maintenance requirements

11.2.2 Production Reliability Degradation Assessment & Control

As was previously shown, the extent of reliability degradation during production depends on the effectiveness of the inspection and quality engineering control program. Reliability analysis methods are applied to measure and evaluate its effectiveness and to determine the need for process improvement or corrective changes. The accomplishment of the analysis task and, more important, how well subsequent corrective measures are designed and implemented will dictate the rate at which reliability degrades/grows during production. Specifically, reliability degradation is minimized during manufacturing, and reliability grows as a result of improvements or corrective changes that:

(1) Reduce process-induced defects through:
- Accelerated manufacturing learning
- Incorporation of improved processes

(2) Increase inspection efficiency through:
- Accelerated inspector learning
- Better inspection procedures
- Incorporation of controlled screening and burn-in tests

As process development and test and inspection efforts progress, problem areas become resolved. As corrective actions are instituted, the outgoing reliability approaches the inherent (design-based) value.

The approach to assessing and controlling reliability degradation involves quantifying process-induced defects and determining the effectiveness of the inspections and tests to remove the defects, i.e., estimating the number of defects induced during assembly and subtracting the number estimated to be removed by the quality/reliability inspections and tests. This includes estimating defects attributable to purchased components and materials, as well as those due to faulty workmanship during assembly.

Process-induced defects can be brought about by inadequate production capability or motivation and from fatigue. Quality control inspections and tests are performed to “weed out” these defects. No inspection process, however, can remove all defects. A certain number of defects will escape the production process, be accepted, and the item released to storage or field operation.

More important, these quality defects can be overshadowed by an unknown number of latent defects. These latent defects, which ordinarily pass factory quality inspection, are due to flaws, either inherent to the parts or induced during fabrication, that weaken the fabricated hardware such that it will fail later under the proper condition of stress during field operation. Reliability screen tests (Environmental Stress Screening) are designed to apply a stress during manufacturing, at a given magnitude and over a specified duration, to identify these latent defects. As in the case of conventional quality inspections, screen tests designed to remove latent defects are not 100% effective.

It must be emphasized that reliability prediction and analysis methods, as discussed in Sections 6, 7, and 8, are based primarily on system design characteristics and data emphasizing the attribute characteristics of the constituent parts. Resulting estimates generally reflect the reliability potential of a system during its useful life period, i.e., during the period after early design when quality defects are dominant and prior to the time when wearout becomes dominant. They represent the inherent reliability, or the reliability potential, of the system as defined by its design configuration, stress and derating factors, application environment, and gross manufacturing and quality factors. A design-based reliability estimate does not represent the expected early life performance of the system, particularly as it is initially manufactured.

11.2.2.1 Factors Contributing to Reliability Degradation During Production: Infant Mortality

In order to assess the reliability of a system or equipment item during its initial life period (as well as during wearout), it is necessary to evaluate the components of failure that comprise its overall life characteristics curve. In general, the total distribution of failure over the life span of a large population of a hardware item can be separated into quality, reliability, wearout and design failures as shown in Table 11.2-3. These failure distributions combine to form the infant mortality, useful life, and wearout periods shown in Figure 11.2-3. It should be noted that design and reliability defects normally would exhibit an initially high but decreasing failure rate and that in an immature design these defects would dominate all other defects.

TABLE 11.2-3: FOUR TYPES OF FAILURES

QUALITY	Unrelated to operating stress	Eliminated by process control and inspection
RELIABILITY	Stress dependent	Minimized by screening
WEAROUT	Time dependent	Eliminated by replacement
DESIGN	May be stress and/or time dependent	Eliminated by proper application, derating, testing and failure data analysis

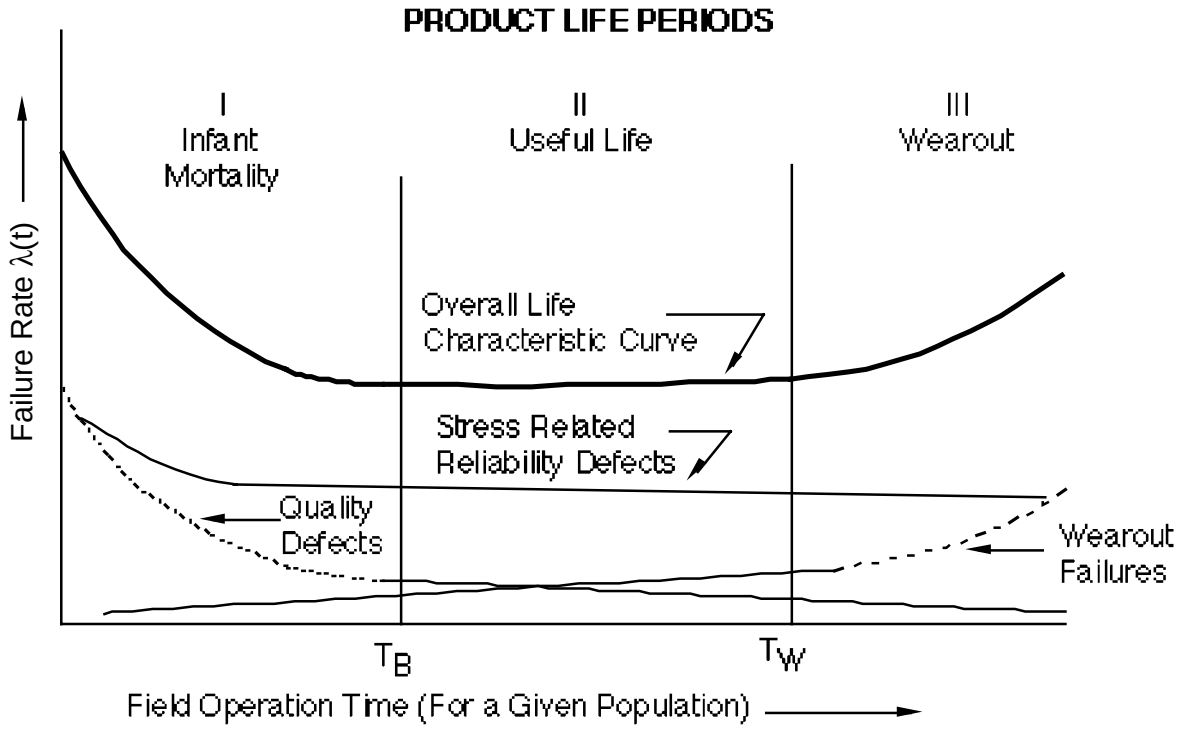


FIGURE 11.2-3: LIFE CHARACTERISTIC CURVE

As indicated in earlier sections, the general approach to reliability design for electronic equipment/systems is to address the useful life period, where failure rate is constant. Design is focused on reducing stress-related failures and generally includes efforts to select high quality, long life parts that are adequately derated.

For new items, this design-based approach in itself is not adequate to ensure reliability. Examination of Figure 11.2-3 shows that the infant mortality period consists of a high but rapidly decreasing quality-related failure distribution, a relatively high and decreasing latent stress-related (reliability) failure distribution, and a low but slightly increasing wearout-related failure distribution. Experience has shown that the infant mortality period can vary from a few hours to well over 1000 hours, although for most well designed, complex equipment it is seldom greater than 100 hours. The duration of this critical phase in reliability growth depends on the maturity of the hardware and, if not controlled, would dominate the overall mortality behavior, leaving the item without a significantly high reliability period of useful life. Positive measures must be taken, beginning with design, to achieve a stabilized low level of mortality (failure rate). This includes evaluating the impact of intrinsic part defects and manufacturing process-induced defects, as well as the efficiency of conventional inspections and the strength of reliability screening tests.

The intrinsic defects arise from the basic limitation of the parts that constitute the system or equipment and are a function of the supplier's process maturity, and inspection and test methods. Intrinsic (or inherent) reliability is calculated using design-based reliability prediction techniques (e.g., MIL-HDBK-217 methods described in Section 6).

The process-induced defects, as previously discussed, are those which enter or are built into the hardware as a result of faulty workmanship or design, process stresses, handling damage, or test efforts and lead to degradation of the inherent design-based reliability. Examples of the types of failures which may occur due to manufacturing deficiencies are poor connections, improper positioning of parts, contamination of surfaces or materials, poor soldering of parts, improper securing of component elements, and bending or deformation of materials.

These defects, as mentioned earlier, whether intrinsic to the parts or introduced during fabrication, can be further isolated into quality and reliability defects. Quality defects are not time dependent and are readily removed by conventional quality control measures (i.e., process control, inspections and tests). The better the process and the more efficient the inspection and test the more defects that are avoided or removed. However, since no test or inspection is perfect, some defects will escape to later manufacturing stages and then must be removed at a much higher cost or, more likely, pass through to field use and thus result in lower actual operational reliability with higher maintenance cost.

Stress/time dependent reliability defects cannot generally be detected (and then removed) by conventional QC inspections. These defects can only be detected by the careful and controlled application of stress screen tests. Screen tests consist of a family of techniques in which electrical, thermal, and mechanical stresses are applied to accelerate the occurrence of potential failures. By this means, latent failure-producing defects, which are not usually detected during normal quality inspection and testing, are removed from the production stream. Included among these tests are temperature burn-in, temperature cycling, vibration, on/off cycling, power cycling, and various nondestructive tests. Burn-in is a specific subclass of screens which employs stress cycling for a specified period of time. A discussion of screening and burn-in is presented in the next section.

As an example of two types of defects, consider a resistor with the leads bent close to its body. If the stress imposed during bending caused the body to chip, this is a quality defect. However, had the stress been inadequate to chip the body, the defect would go unnoticed by conventional inspection. When the body is cycled through a temperature range, small cracks can develop in the body. This would allow moisture and other gases to contaminate the resistive element, causing resistance changes. This is a reliability defect. Note that this defect can also be a design defect if the design specifications require a tight bend to fit the component properly in a board. However, if the improper bend is due to poor workmanship or process design, the defect is classified as a process-induced reliability defect. Consequently, the types of defects to which a system and its subsystems are susceptible are determined by the parts selected and their processing, while the presence of these defects in the finished item is a function of the quality

controls, tests and screens that are applied.

Figure 11.2-4 pictorially shows the reliability impact of the part and process defects. As shown, an upper limit of reliability is established by design based on part derating factors, application environment, quality level, etc. The shaded area indicates that the estimated inherent reliability level may have a relatively broad range depending on the parts that comprise the system and the values for the parameters of the part failure estimating models.

The reliability of initially manufactured units will then be degraded from this upper limit; subsequent improvement and growth is achieved through quality inspections, reliability screening, failure analysis, and corrective action. The extent and rigor with which the tests, failure analysis and corrective actions are performed determine the slope of the reliability improvement curve. As such, process defects, along with the inherent part estimates, must be evaluated in order to accurately estimate reliability, particularly during initial manufacturing.

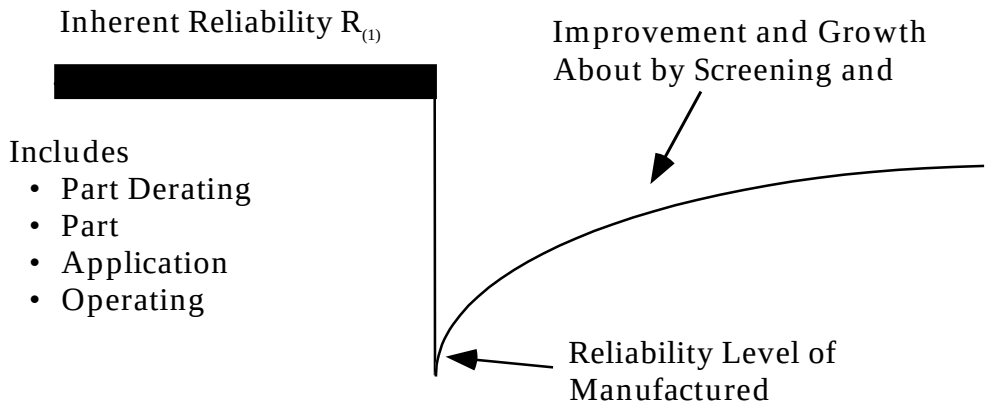


FIGURE 11.2-4: IMPACT OF DESIGN AND PRODUCTION ACTIVITIES ON EQUIPMENT RELIABILITY

11.2.2.2 Process Reliability Analysis

The infant mortality period (as was shown in Figure 11.2-3) is composed of a high but rapidly-decreasing quality component, a relatively high and decreasing stress component, and a low but slightly increasing wearout component. Because of this non-constant failure rate, this life period cannot be described simply by the single parameter exponential distribution; computation of reliability during this period is complex. It would require application of the Weibull distribution or some other multi-parameter distribution to account for the decreasing failure rate. Controlled life tests would have to be performed or extensive data compiled and statistically evaluated to determine the parameters of the distributions.

A practical approach, however, that would perhaps be useful during pre-production planning or during early production is to compute an average constant failure rate (or MTBF). This average MTBF represents a first approximation of the reliability during this early period. It can be viewed as a form of "step" MTBF, as shown in Figure 11.2-5 where the "step" MTBF includes both stress and quality failures (defects) at both the part and higher assembly levels, while the inherent MTBF (experienced during the useful life period) includes only stress related failure (defects) at the part level.

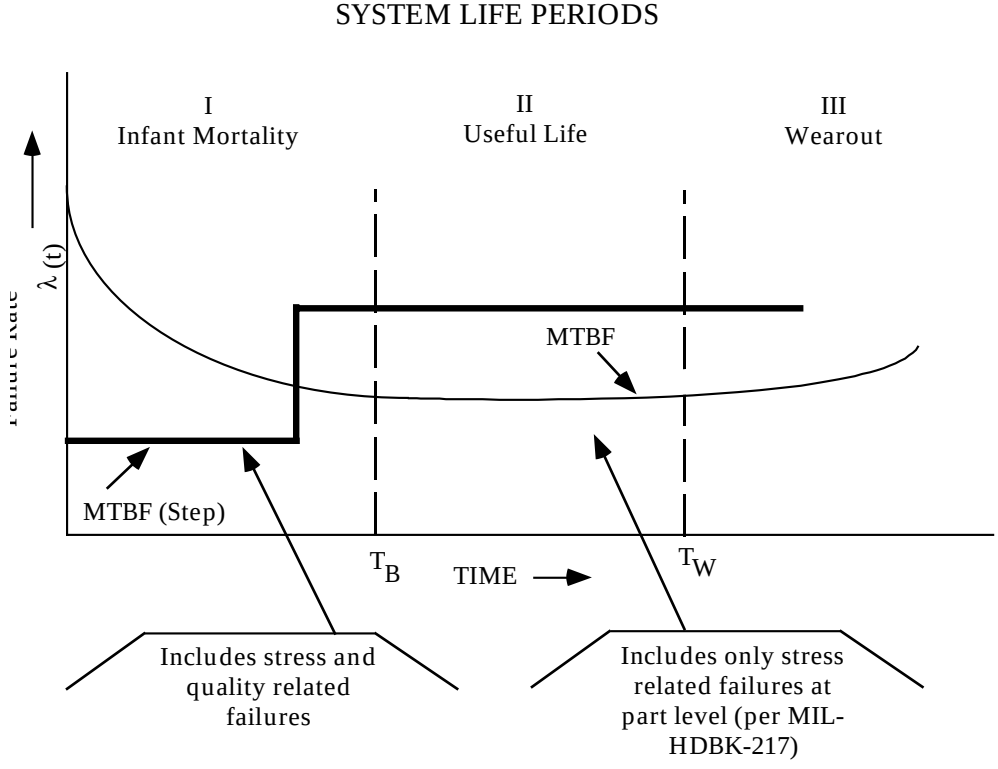


FIGURE 11.2-5: "STEP" MTBF APPROXIMATION

A production reliability and inspection analysis can be performed to compute this average "step" MTBF. Such an analysis, in its simplest form, will determine where large degrees of unreliability (defects) are introduced in the manufacturing process and, thus, provides a basis to formulate and implement corrective action in response to the overall improvement process.

This "step" MTBF or outgoing from production MTBF (initial manufacturing) is computed from the following expression:

$$MTBF_a = MTBF_i \, D_k \qquad (11.1)$$

where:

$MTBF_a$ = initial manufacturing MTBF

$MTBF_i$ = the inherent MTBF and is computed from part failure rate models as described in Section 6

D_k = overall degradation factor due to effects of process and inspection efficiency

$$Dk = D_i/D_{out} \qquad (11.2)$$

where:

D_i = the inherent probability of defects that is computed from $MTBF_i$, i.e,

$$D_i = 1 - e^{-t/MTBF_i}$$

and

$$MTBF_i = 1/\lambda_i$$

$$\lambda_i = (\lambda_{OP})\, d + (\lambda_{NON\text{-}OP})\,(1\text{-}d)$$

λ_{OP} = operational failure rate

d = ratio of operational time to total time

NON-OP = failure rate for non-operational time

Nonoperational failure rates ($\lambda_{NON\text{-}OP}$) have been traditionally calculated by one of two methods:

(1) Multiplicative "K" factor applied to operating λ

(2) Operating failure rate model extrapolated to zero stress

To provide a more accurate method for defining the nonoperating failure rate of electronic equipment, Rome Air Development Center published a report (RADC-TR-85-91) entitled *"Impact of Nonoperating Periods on Equipment Reliability"* (Ref. [15]).

The objective of this study was to develop a procedure to predict the quantitative effects of nonoperating periods on electronic equipment reliability. A series of nonoperating failure rate prediction models consistent with current MIL-HDBK-217 models were developed at the component level. The models are capable of evaluating component nonoperating failure rate for any anticipated environment with the exception of a satellite environment.

This nonoperating failure rate prediction methodology provides the ability to predict the component nonoperating failure rate and reliability as a function of the characteristics of the device, technology employed in producing the device, and external factors such as environmental stresses which have a significant effect on device nonoperating reliability. An analytical approach using observed data was taken for model development where possible. Thus, the proposed models only include variables which can be shown to significantly affect nonoperating failure rate. The prediction methodology is presented in a form compatible with MIL-HDBK-217 in Appendix A of the report.

Use of a multiplicative “K” factor has merit under certain circumstances. The “K” factor can be accurately used to predict nonoperating failure rate if it was based on equipment level data from the same contractor on a similar equipment type with similar derating and screening. In any other circumstances, the use of a “K” factor is very approximate method at best. Additionally, it is intuitively wrong to assume that operating and nonoperating failure rates are directly proportional. Many application and design variables would be anticipated to have a pronounced effect on operating failure rate, yet negligible effect on nonoperating failure rate. Derating is one example. It has been observed that derating results in a significant decrease in operating failure rate, but a similar decrease would not be expected with no power applied. Additionally, the stresses on parts are different in the nonoperating state, and therefore, there is no reason to believe that the operating factors for temperature, environment, quality and application would also be applicable for nonoperating reliability prediction purposes.

An invalid approach for nonoperating failure rate assessment has been to extrapolate operating failure rate relationships to zero electrical stress. All factors in MIL-HDBK-217, whether for electrical stress, temperature or another factor, represent empirical relationships. An empirical relationship is based on observed data, and proposed because of the supposedly good fit to the data. However, empirical relationships may not be valid beyond the range of parameters found in the data and this range does not include zero electrical stress for MIL-HDBK-217 operating reliability relationships. Extrapolation of empirical relationships beyond the range found in the data can be particularly dangerous when the variable is part of an exponential relationship. A relatively small error in the exponent can correspond to a large error in the resultant predicted failure rate. Additionally, there are many intuitive or qualitative reasons why small amounts of applied power can be preferable to pure storage conditions. For nonhermetic microcircuits, the

effect of humidity is the primary failure accelerating stress. A small current will result in a temperature rise, burning off moisture, and probably decreasing device failure rate.
Figure 11.2-6 depicts the steps involved in performing a complete reliability analysis leading to an average MTBF ($MTBF_a$) for the early production period of a new hardware item as well as the MTBF ($MTBF_i$) during its useful life period. The analysis involves first evaluating the item's design to determine the inherent (design based) $MTBF_i$. Once the design analysis is completed, the process and inspection analysis is performed, as discussed previously, to determine the outgoing (from production) defect rate, D_{out}, and, ultimately, the factor D_k that accounts for degradation in reliability due to initial manufacturing. The output of these two efforts is then combined to yield an MTBF estimate that would account for initial manufacturing induced defects.

The analysis, as depicted in Figure 11.2-6, involves the following steps:

Step 1: Compute the reliability of the system or equipment item as it enters the manufacturing process. The initial estimate of reliability is based upon inherent $MTBF_i$ prediction as previously discussed.

Step 2: Construct a process and inspection flow diagram. The construction of such a flow chart involves first the identification of the various processes, inspection, and tests which take place during manufacturing and second a pictorial presentation describing how each process flows into the next process or inspection point. Figure 11.2-7 presents a basic and highly simplified process flow diagram to illustrate the technique. Since the analysis may be performed on new equipment prior to production or equipment during production, the process diagram may depict either the planned process or the existing production process.

Step 3: Establish reject rate data associated with each inspection and test. For analysis performed on planned processes, experience factors are used to estimate the reject rates. The estimated reject rates must take into account historical part/assembly failure modes in light of the characteristics of the test to detect that failure mode. Some of the tests that are used to detect and screen process-induced defects and which aid in this evaluation are discussed in the next section. For analysis performed on existing production processes, actual inspection reject rate data can be collected and utilized.

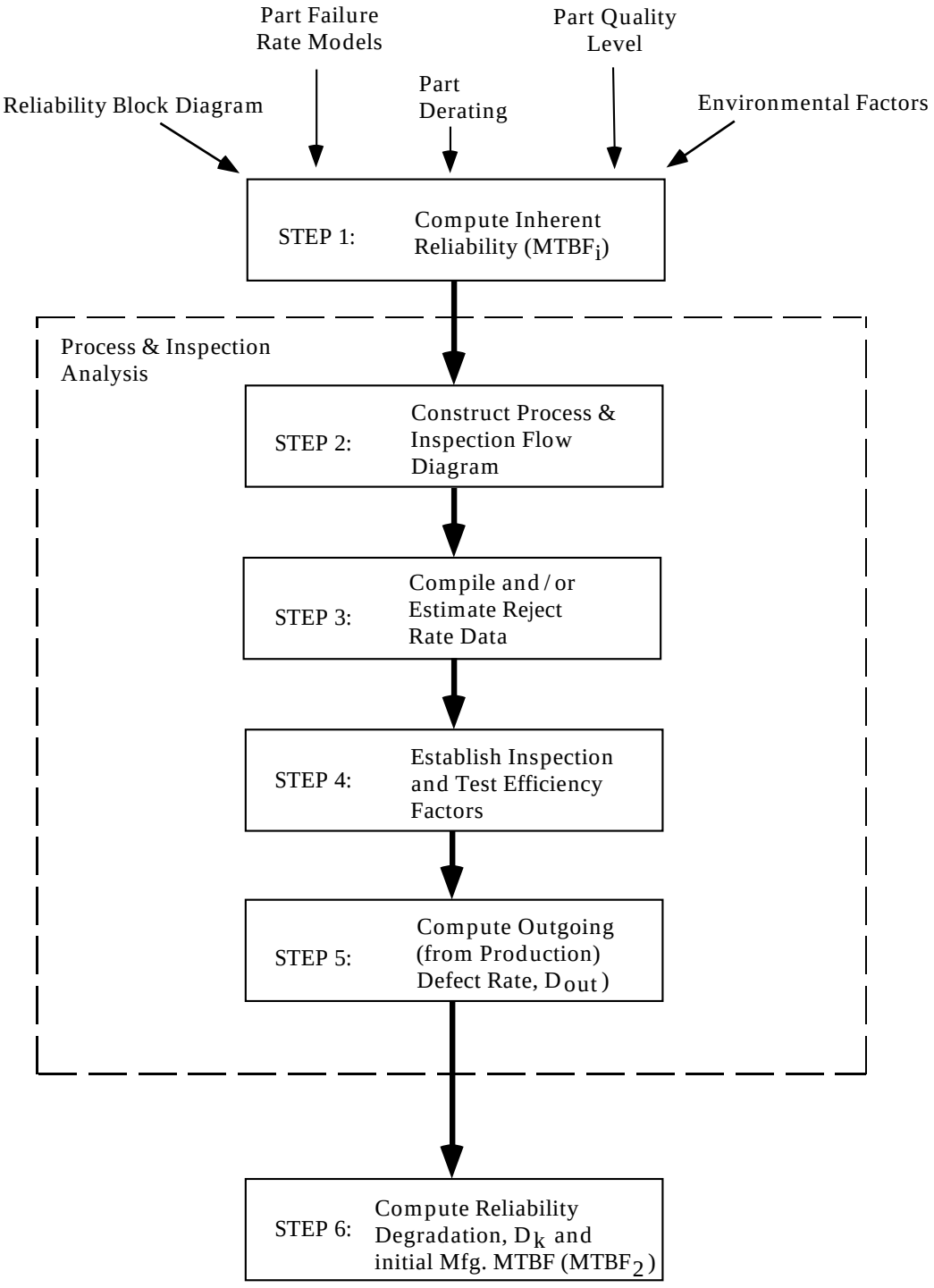


FIGURE 11.2-6: MTBF (OUTGOING FROM PRODUCTION) ESTIMATING PROCESS

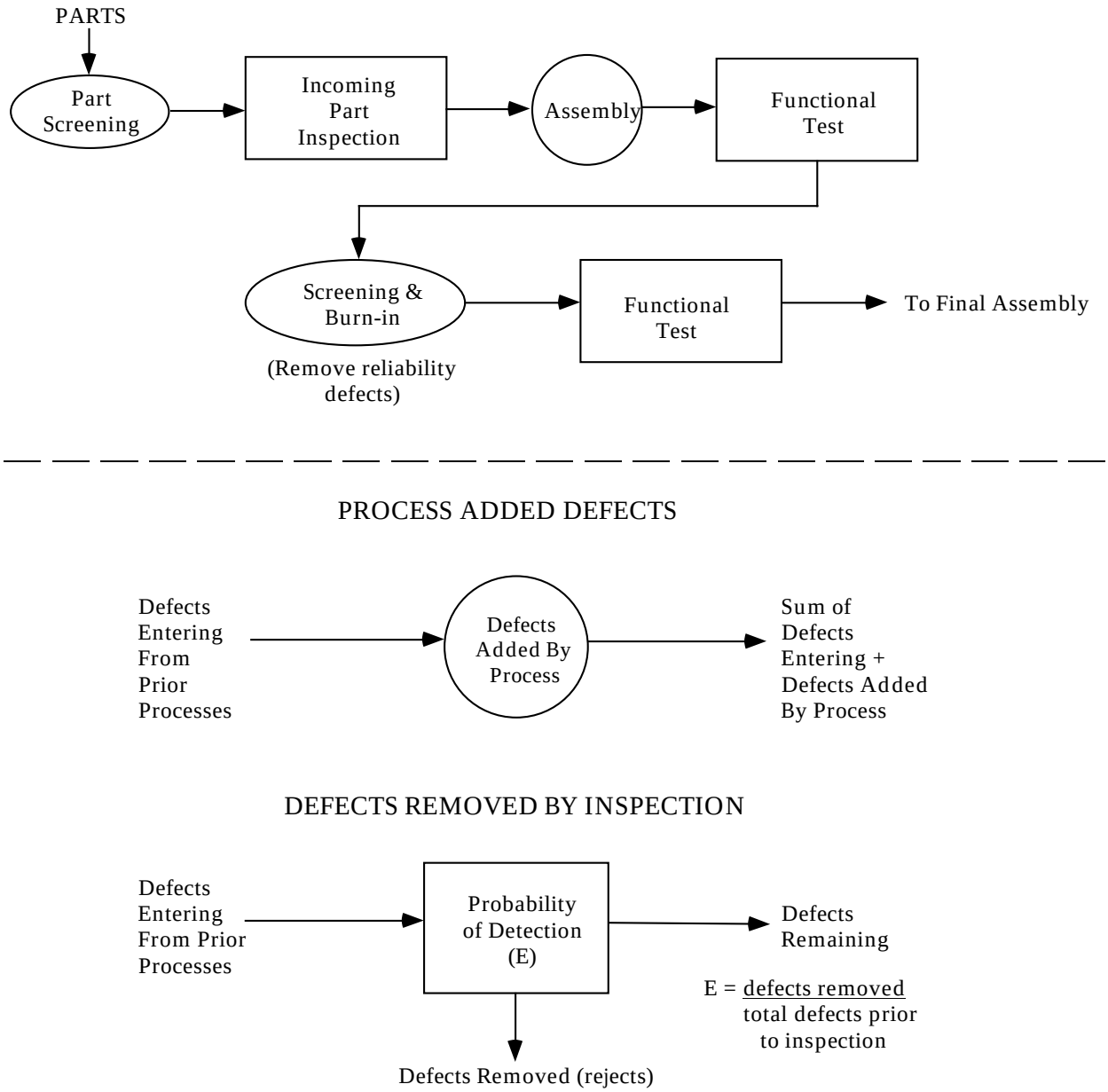


FIGURE 11.2-7: SAMPLE PROCESS FLOW DIAGRAM

Step 4: Establish inspection and test efficiency factors. Efficiency is defined as the ratio of defects removed (or rejects) to the total defects in the fabricated items. Efficiency factors are based on past experience for the same or a similar process, when such data exists. For newly instituted or proposed inspection and screen tests having little or no prior history as to how many defects are found, estimates of inspection and test efficiency must be made. To estimate efficiency, the inspections can be described and characterized relative to such attributes as:

(1) Complexity of part/assembly under test
(e.g., simple part, easy access to measurement)

(2) Measurement equipment
(e.g., ohmmeter for short/open circuit check, visual for component alignment check)

(3) Inspector experience
(e.g., highly qualified, several years in quality control)

(4) Time for inspection
(e.g., production rate allows adequate time for high efficiency)

(5) Sampling plan
(e.g., 100% of all parts are inspected)

Weight factors can be applied to each of the inspection attributes and used to estimate percent efficiency.

Step 5: Compute outgoing defect rate based on the reject rates (from Step 3) and the efficiency factors (Step 4) using the process flow diagram developed during Step 2. Note that for a given inspection with a predefined efficiency factor, E, the number of defects of a fabricated item prior to its inspection can be estimated from the measured or estimated rejects, i.e., E = number rejected/total defects (prior to inspection). The number of outgoing defects is simply the difference between the number prior to inspection and that removed by the inspection.

Step 6: Compute reliability degradation based on the ratio of the inherent design based reliability (Step 1) and the outgoing-from-manufacturing defect rates (Step 5). Note: Not all defects result in an actual hardware failure. Though a defect may exist, it may not be stressed to the point of failure. Through the reduction of the outgoing defect rates for a production process, field defect rates are reduced and, thus, reliability is improved.

Hardware reliability can be improved through successive application of the above

analysis. Those processes, wherein large numbers of defects are being introduced, can be isolated and corrected or changed with an improved process or by applying a screen test (or sequence of tests) to remove the defects. The inclusion of a screening test will increase the initial cost of the system, but the cost avoidance due to increased factory productivity (i.e., lower rework, scrap rate, etc.) and, more important, the lower field maintenance and logistics support cost, should more than offset the initial cost. To be most cost-effective, particularly for large complex systems, the application of the production reliability and inspection analysis should be first applied to subsystems and equipment designated as critical by methods such as the failure mode and effects analysis procedures described in Section 6.

11.2.3 Application of Environmental Stress Screening (ESS) during Production to Reduce Degradation and Promote Growth

A clear understanding of environmental stress screening (ESS) requires a good definition as a baseline. The following definition addresses the key aspects of ESS:

Environmental stress screening of a product is a process which involves the application of one or more specific types of environmental stresses for the purpose of precipitating to hard failure, latent, intermittent, or incipient defects or flaws which would otherwise cause product failure in the use environment. The stresses may be applied either in combination or in sequence on an accelerated basis but within product design capabilities.

One of the keystones of an effective production reliability/assessment and control program is the proper use of screening procedures. ESS is a procedure, or a series of procedures, specifically designed to identify weak parts, workmanship defects and other conformance anomalies so that they can be removed from the equipment prior to delivery. It may be applied to parts or components, boards, subassemblies, assemblies, or equipment (as appropriate and cost effective), to remove defects which would otherwise cause failures during higher-level testing or during early field operation. ESS is described in detail in the reliability testing specification MIL-HDBK-781, Task 401, *"Environmental Stress Screening (ESS)"*.

Historically the government explicitly specified the screens and screening parameters to be used at various assembly levels. Failure-free periods were sometimes attached to these screens, as an acceptance requirement, in order to provide assurance that the product is reasonably free of defects. This approach is documented in MIL-HDBK-2164A, *"Environmental Stress Screening Process for Electronic Equipment."*

Under acquisition reform, the government refrains from telling contractors "how" to do things. With this philosophy, the contractor develops and proposes an environmental stress screening program for the equipment which is tailored to the product. MIL-HDBK-344, *"Environmental*

Stress Screening (ESS) of Electronic Equipment," provides guidelines to assist the contractor in the development and establishment of an effective ESS program. It establishes a set of procedures and ground rules for the selection of the proper type of stress, the amount of stress, and the duration of the stress or stresses to be used in the formulation of a cost-effective environmental stress screening program for a specific equipment. It also describes general techniques for planning and evaluating ESS programs.

Additional guidance on ESS can be found in "Environmental Stress Screening Guidelines for Assemblies," dated 1990, from the Institute of Environmental Sciences (Ref. [13]) and the Tri-Service Technical Brief 002-93-08, *"Environmental Stress Screening (ESS) Guidelines,"* dated July 1993 (Ref. [11]).

The purpose of ESS is to compress a system's early mortality period and reduce its failure rate to acceptable levels as quickly as possible. The rigor of the applied stresses and subsequent failure analysis and corrective action efforts determines the extent of degradation in reliability as well as the degree of improvement. A thorough knowledge of the hardware to be screened and the effectiveness and limitations of the various available screenings is necessary to plan and implement an optimized production screening program.

Screening generally involves the application of stress during hardware production on a 100 percent basis for the purpose of revealing inherent, as well as workmanship and process-induced, defects without weakening or destroying the product. The application of stress serves to reveal defects which ordinarily would not be apparent during normal quality inspection and testing. There are a large number of stresses and stress sequences that can be applied to reveal defects induced at the various levels of fabricated assembly. Each specific screening program must be designed and optimized relative to the individual hardware technology, complexity, and end item application characteristics, as well as the production volume and cost constraints of the product being manufactured. Planning a screening program is an iterative process that involves tradeoff analysis to define the most cost-effective program.

Screening can be applied at the part, assembly, and system levels. In order to detect and eliminate most of the intrinsic part defects, initial screening may be conducted at the part level. Certain defects, however, are more easily detected as part of an assembly test. This is particularly true of drift measurements and marginal propagation delay problems. Assembly defects, such as cold solder joints, missing solder joints and connector contact defects can be detected only at the assembly or subsystem level. At higher assembly levels, the unit's tolerance for stress is lower and, thus, the stress that can be safely applied is lower. As a general rule, screens for known latent defects should be performed as early in the assembly process as possible. They are most cost effective at this stage. A standard rule of thumb used in most system designs is that the cost of fixing a defect (or failure) rises by an order of magnitude with each assembly level at which it is found. For example, if it costs x dollars to replace a defective part, it will cost 10x to replace that part if the defect is found at the printed circuit board level, 100x if found at the equipment level, etc.

Figure 11.2-8 depicts a typical production process where parts and printed circuit boards (PCBs) or wired chassis comprise assemblies; then manufactured assemblies, purchased assemblies and associated wiring comprise units; and finally the units, other equipment and intercabling make up the completed system. Latent defects are introduced at each stage in the process and, if not eliminated, propagate through to field use. The cost of repair increases with increasing levels of assembly, being $6 to $25 at the part level and perhaps as high as $1500 at the system level. Field repair cost estimates have been quoted as high as $20,000. This data would tend to validate the previously mentioned rule of thumb. Thus, for economic reasons alone, it is desirable to eliminate latent defects at the lowest possible level of assembly, and certainly, prior to field use.

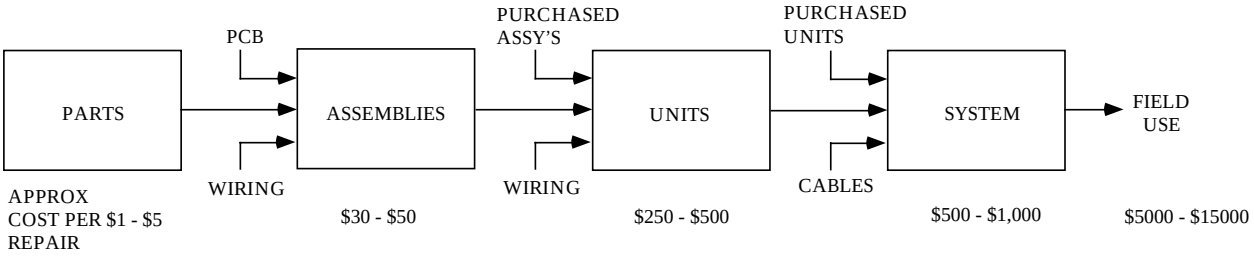


FIGURE 11.2-8: A TYPICAL PRODUCTION PROCESS, FINDING DEFECTS AT THE LOWEST LEVEL OF MANUFACTURE IS THE MOST COST-EFFECTIVE

The idealized manufacturing process, depicted in Figure 11.2-9, starts with screened parts procured and received to a predetermined level of quality.

Screening is then applied as required at the different levels of assembly. All screening rejects are analyzed. The results of this analysis are used to identify appropriate product design changes and modifications to the manufacturing process, and to reduce, if possible, the overall test burden. All screening results, including reject rates, failure modes, and time-to-failure data are incorporated into a dynamic real-time database by which the effectiveness of the screening program is continuously assessed. The database also represents a primary experience pool for designing new screening programs as new systems are developed and introduced into the manufacturing stream.

Screening can be applied at the three major levels of assembly: part, intermediate (i.e., PCB), and unit/equipment or system. Initial planning and tradeoff studies should take into account the effectiveness and the economic choices between part, intermediate, and final equipment/system level screens and their applicable parameters.

11.2.3.1 Part Level Screening

Part level screening is relatively economical and can be incorporated into supplier specifications where necessary. It has the potential for maximum cost avoidance, particularly when applied to low volume parts, such as, complex hybrid microcircuits and other high technology devices

where reliability is highly dependent upon specific fabrication techniques and very explicit process controls. Screen stress levels can be matched to requirements, which, in general, enable the safe application of higher and more effective stress levels to remove known part defects. Part level screens offer procedural simplicity and the ability to pass a great deal of the burden for corrective action back to the part vendors. Low level screens, however, have no impact on the control of defects introduced during subsequent phases of manufacture and assembly.

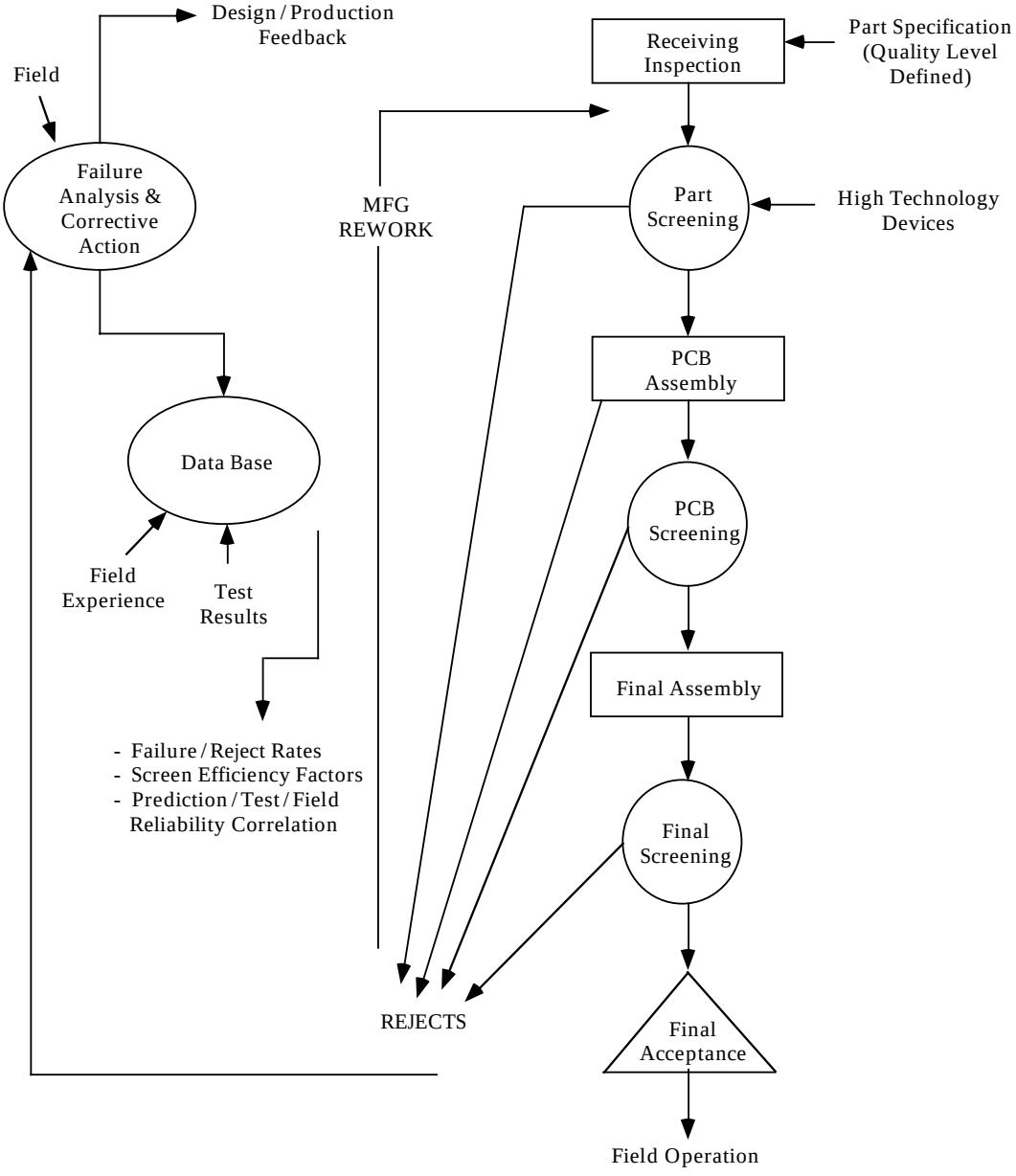


FIGURE 11.2-9: APPLICATION OF SCREENING WITHIN THE MANUFACTURING PROCESS

With the significant growth in the commercial market, accompanied by the decline in the military market, commercial microcircuits are being used more and more frequently in military equipments today. These parts are generally not subjected to special manufacturing procedures, inspections, or burn-in but, as a minimum, generally undergo some form of visual and electrical parameter screening. The normal Statistical Process Control (SPC) procedures incorporated in the continuous high volume production of commercial microcircuits is generally sufficient to assure the quality of these devices. A problem arises, however, when the device volume is not sufficiently large to assure the effectiveness of the manufacturer's SPC or where SPC is not utilized effectively or not at all. Then part level screening becomes a viable alternative and indeed a necessity for some specific parts.

Screening and inspection tests for resistors, capacitors and other passive components typically include high temperature conditioning, visual and mechanical inspections, dc resistance measurement, low temperature operation, temperature cycling, moisture resistance, short time overload, shock, vibration, solderability, and rated power life test.

11.2.3.2 Screening at Higher Levels of Assembly

Among military electronic equipment manufacturers, environmental stress screening at the module and equipment level has increased significantly in recent years. The general consensus is that temperature cycling is the most effective stress screen, followed by random vibration (Ref. [2]) as shown in Figure 11.2-10.

The Institute of Environmental Sciences (IES), a professional organization of engineers and scientists, has developed a guidelines document (Ref. [2]) for Environmental Stress Screening of Electronic Hardware (ESSEH).

Intermediate screening is more expensive but can remove defects introduced at the board level as well as those intrinsic to the parts. Because of the several part types incorporated into a board, somewhat lower stress levels must be applied. For example, the maximum screening temperature depends upon the part with the lowest maximum temperature rating of all the parts on the board. Generally, special burn-in/temperature cycling facilities are required as well as special automatic test equipment (ATE). In general, some amount of ATE is employed in virtually all large scale screening programs. Automatic testing cannot only perform rapid functional testing after screening of complex boards (or other assemblies) but also is effective in the detection of pervasive faults. The latter consist of marginal performance timing problems and other defects arising from part interactions during operation. The extent of the facilities and equipment needed is dependent on the test conditions specified. The potential for cost avoidance with intermediate level screens is not as high as for part level screens, and the necessity to employ, generally, a lower stress level reduces their effectiveness to some extent.

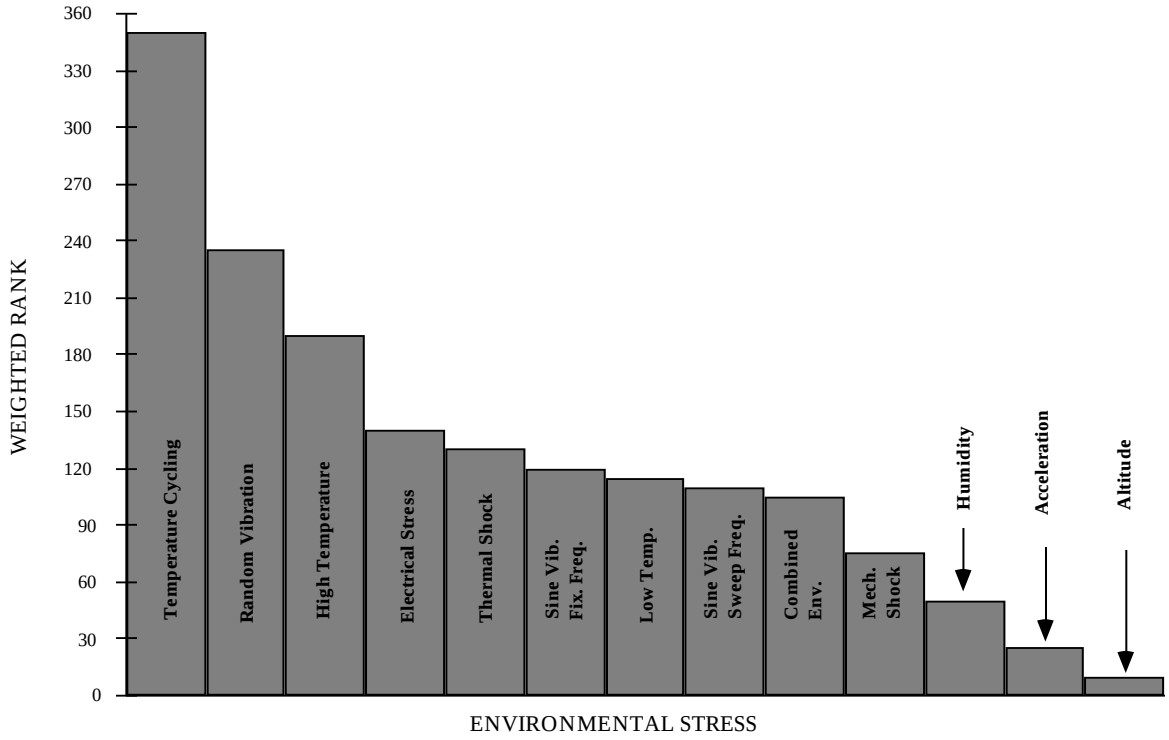


FIGURE 11.2-10: EFFECTIVENESS OF ENVIRONMENTAL SCREENS

Equipment/system level screening is expensive but it can remove defects introduced at all levels of fabrication. At this point in the manufacturing stream, the potential for cost avoidance is low and the permissible stress level may not adequately exercise certain specific parts. However, higher level assembly tests are considered important, even if it is thought that the lower level tests may have eliminated all defective parts and board defects. The process of assembling the remaining components and the boards into the larger assemblies and into the final item cannot be assumed to be free of failure- producing defects. Good parts may be damaged in final assembly, workmanship errors can occur, and product-level design defects may be present.

Unit/equipment screens are primarily intended to precipitate unit workmanship defects and, secondarily, assembly level defect escapes. Unit level defects vary with construction but typically include interconnection defects such as:

(1) PWB connector (loose, bent, cracked or contaminated contacts, cracked connector)

(2) Backplane wiring (loose connections, bent pins, damaged wire insulation, debris in wiring)

(3) Unit input/output connectors (loose, cracked pins, damaged connector, excessive, inadequate or no solder on wire terminations, inadequate wire stress relief)

(4) Intra-unit cabling (improperly assembled coax connectors, damaged insulation)

Units may also contain wired assemblies integral to the unit and not previously screened, such as Power Control and BIT Panels, and purchased assemblies, such as modular power supplies.

11.2.3.3 Screen Test Planning and Effectiveness

An effective reliability screening program requires careful planning that starts during early development. Tradeoff studies are performed and a complete test specification is prepared and possibly verified for its effectiveness on prototype hardware.

A key step in specifying an effective screening program is the identification of the kinds of failure modes that can occur and the assembly level at which they may be induced. The appropriate screens are those which are most effective in accelerating the identified modes, whether they are intrinsic to the part or induced by the manufacturing process.

Due to the varied nature of military electronics equipments and their associated design, development and production program elements, it is difficult to "standardize" on a particular screening approach. A tailoring of the screening process to the unique elements of a given program is, therefore, required.

Screens should be selected based upon estimates of cost and effectiveness, early development program data, equipment design, manufacturing, material and process variables, which at least narrow consideration to the most cost effective choices. The screening process then should be continuously monitored and the results analyzed so that changes in the process can be made as required to optimize the cost effectiveness of the screening program.

11.2.3.3.1 Environmental Stress Screening per MIL-HDBK-344

MIL-HDBK-344 is organized according to the general sequence of events for planning, monitoring and controlling a screening program. Five detailed procedures are used to assist the user in accomplishing ESS planning and evaluation activities. The detailed procedures are briefly described as follows:

Procedure A - Optimizing Screen Selection and Placement

This procedure is used to plan an ESS program such that the required field reliability is attained at an optimum combined user-producer cost. Procedures B and C are subsequently used to design the ESS program, and Procedure D is then used to validate the original estimates of defect density and screening strength and to redefine the ESS program as necessary.

Five detailed procedures are contained within Procedure A. Procedure A1 creates the basic ESS model for a particular program and determines the incoming defect density, allowable outgoing

defect density based on a reliability requirement, and factory ESS constraints. Procedure A2 optimizes the combined user/producer cost of achieving the specified reliability using the results of Procedure A3. Procedure A3 calculates the cost of the ESS program. Procedure A4 is used as a precautionary measure to ensure that the ESS is not too stressful and does not consume too much of the useful (fatigue) life. Procedure A5 is then used to refine the program, as designed using procedures A1 through A4, by determining actual values for incomming defects (D_{IN}), screening strength (SS), detection efficiency (DE), and stress adjustment factor (SAF) from factory and field data analyzed using Procedure D.

Procedure B - Estimating Defect Density

This procedure is used to estimate the number of defects resident in the system prior to beginning ESS. In this procedure the number of defects is defined relative to a baseline stress level. Appropriate factors are then applied to determine the number of defects for different stress levels of vibration, temperature and temperature transition rates that occur in the factory and the field. It is important to address these stress adjustment factors when planning an ESS program since they affect the economic optimization.

The procedure steps are: 1) estimate defects for each assembly and the total system at baseline stress, 2) proportion the defects into random vibration (RV) and temperature cycling (TC) sensitive populations, and 3) apply stress adjustment factors to determine the defects under different factory stress levels. Two procedures are contained within Procedure B. Procedure B1 determines the number of latent defects resident in the equipment at the baseline stress. Procedure B2 determines the stress adjustment factor relating defects at factory (baseline stress) levels to defects at the field application stress levels.

Procedure C - Estimating Screening Strength (SS)

This procedure is used to estimate the number of flaws precipitated and detected (removed) by ESS. Screening strength is characterized by a precipitation term and a detection term and determines the fraction of existing flaws that are removed by ESS. Precipitation is defined as the conversion of a flaw with some residual strength into a flaw with no strength. The application of stress precipitates a certain fraction of the existing flaws. This fraction is assumed to be constant for a specific stress level and duration. The removal of a potential defect or flaw requires the flaw to be precipitated and subsequently detected and removed. Detection efficiency is defined as the capability of detecting, isolating and removing the defect once it has precipitated. Precipitation and detection terms are estimated separately and their product determines the screening strength.

Procedure D - Refining Estimates of Defect Density and Screening Strength

This procedure is used to provide revised estimates of the ESS modeling parameters (D_{IN}, precipitation efficiency (PE) and SS, $D_{REMAINING}$, etc.) using actual factory and field data. The most important parameter for ESS is the defects remaining at the time of shipment since this determines the field reliability. Other significant parameters are the initial defect density, and the screening strength of the various screens. The difficulty, however, is that none of these parameters are directly observable by the producer. Only the defects removed through factory ESS can be measured. This procedure provides the means for determining these other critical parameters from factory data.

Procedure E - Monitor and Control

This procedure is used to implement a program to monitor and control the ESS program (consistent with TQM philosophy) thereby ensuring that the program remains cost effective under the evolving conditions. It provides a quantitative assessment of whether reliability requirements are being attained and to what extent continuous improvement is being realized. The parameters of interest for monitor and control are those determined in Procedure D. Modified SPC and Pareto charts are prepared to monitor these parameters against the requirements which were established in Procedure A.

Procedure F - Product Reliability Verification Test (PRVT)

This procedure is used in conjunction with Procedure E for monitor and control purposes to provide confidence that field reliability will be achieved. The objective is to retain a minimum ESS program so that field reliability can be projected and out-of-control conditions identified. PRVT is defined as that portion of a minimal ESS retained for the purpose of providing a mechanism to indicate when the process is not in control and is an inherent part of the ESS program.

The product development phase is used to experiment with stress screens, and to then define and plan a cost effective screening program for production. After the screening program is implemented during production, stress screening results are used to evaluate the screening process to establish whether program objectives are being achieved.

Quantitative objectives for the screening program must be established early. Appendix A of MIL-HDBK-344 contains the mathematical relations and model descriptions used in the Handbook. A review of Appendix A will help in gaining a quick understanding of the rationale and methodology of the Handbook. A typical task sequence in Planning, Monitoring and Controlling an ESS Program in accordance with MIL-HDBK-344 is shown in Figure 11.2-11.

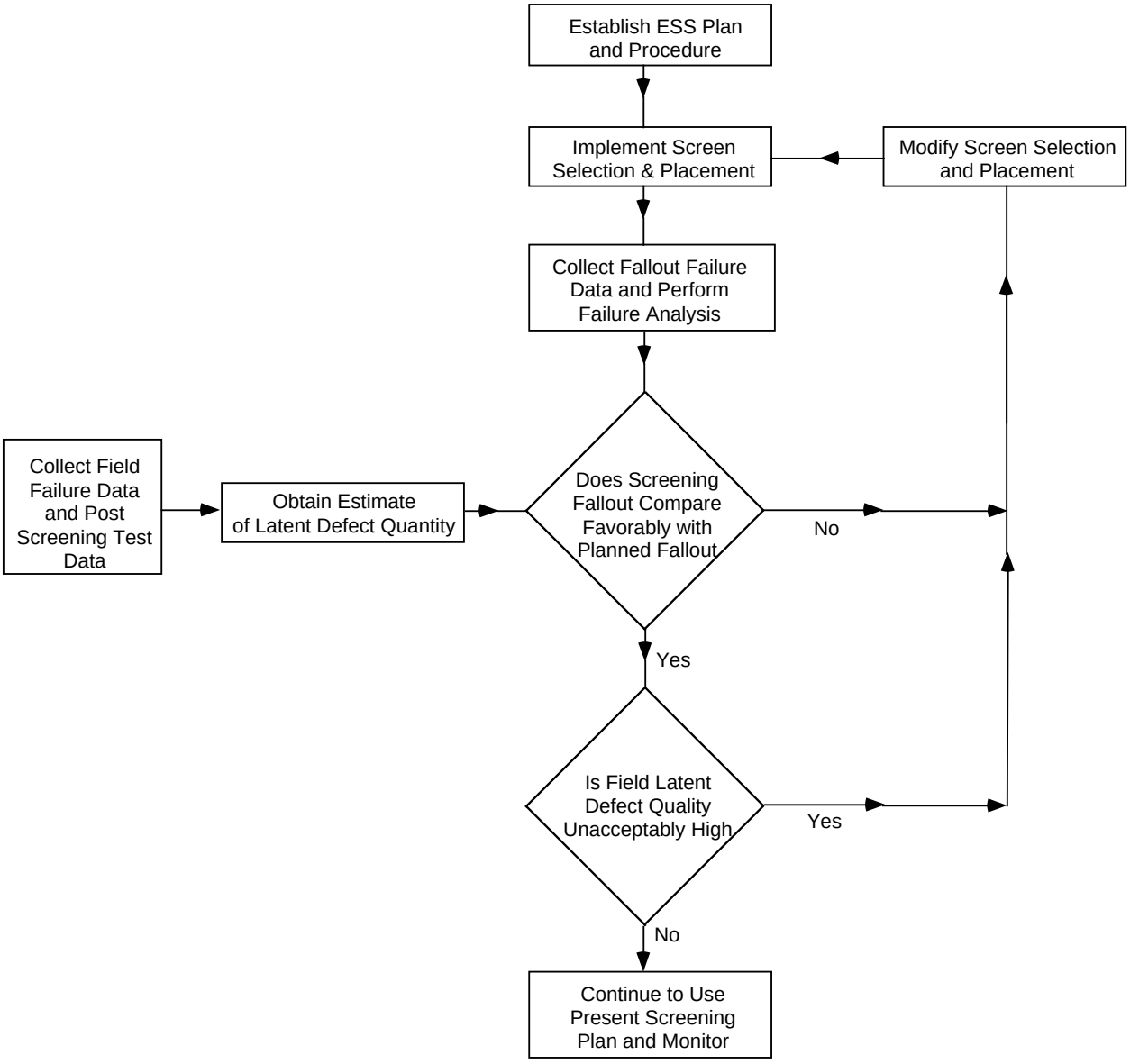


FIGURE 11.2-11: MIL-HDBK-344 ESS PROCESS

11.2.3.3.2 Tri-Service ESS Guidelines

A Tri-Service Technical Brief 002-93-08, Environmental Stress Screening Guidelines, was issued in July 1993. The following excerpts from this document provide examples of its technical guidance and flexibility.

A viable ESS program must be actively managed, and tailored to the particular characteristics of the equipment being screened. A survey should be conducted to determine the mechanical and thermal characteristics of the equipment and refining the screening profiles as more information becomes available and/as designs, processes, and circumstances evolve.

Initially, ESS should be applied to all the units manufactured, including repaired units. By using a closed loop feedback system, information can be collected to eventually determine if the screening program should be modified.

The following summarizes the ESS guidance:

- Define contractual requirements
- Identify a general approach
- Identify the nature of anticipated defects for unit design and manufacturing processes
- Exercise a cost model considering:
  - Assembly level at which to apply ESS
  - Level of automation versus manual labor
  - Specific rates of thermal change versus capital investment to achieve it
  - Adequacy of available in-house random vibration equipment versus cost of off-site screening or the purchase of new equipment
  - Cost considerations of active versus passive screening
- Review available government and industry data relative to the design of screening profiles for comparable equipment
- Review product design information to identify any thermal characteristics or mechanical resonances/weakness which could affect screening profiles
- Tailor and finalize the temperature cycling screen, at each level of assembly selected, for temperature limits, rate of temperature change, number of temperature cycles, and whether monitored during screen
- Tailor and finalize the random vibration screen, at each level of assembly selected, for spectrum, grms level, number of axes, true random or quasi-random, and whether monitored during screen
- Optimize or modify the ESS profiles based on data from the screens or from operational use
- Consider sampling for the ESS screen based on screening data collected, but only with customer concurrence.

11.2.3.3.2.1 Types of Flaws to be Precipitated

Based on the types of failures expected in the equipment, Table 11.2-4 provides information that can be used to help establish the unique equipment profile. Care must be taken that in tailoring the screening environment to one type of failure, other types of failures do not go undetected.

TABLE 11.2-4: SCREENING ENVIRONMENTS VERSUS TYPICAL FAILURE MECHANICS

SCREENING ENVIRONMENT - TYPE OF FAILURE		
THERMAL CYCLING	VIBRATION	THERMAL OR VIBRATION
• Component parameter drift	• Particle contamination	• Defective solder joints
• PCB opens/shorts	• Chaffed, pinched wires	• Loose hardware
• Component incorrectly installed	• Defective crystals	• Defective components
• Wrong component	• Mixed assemblies	• Fasteners
• Hermetic seal failure	• Adjacent boards rubbing	• Broken component
• Chemical contamination	• Two components shorting	• Improperly etched PCBs
• Defective harness termination	• Improperly seated connectors	• Surface mount technology flaws
• Improper crimp	• Poorly bonded component	
• Poor bonding	• Inadequately secured parts	
• Hairline cracks in parts	• Mechanical flaws	
• Out-of-tolerance parts	• Loose wire	

11.2.3.3.2.2 Levels of Assembly at which ESS may be Performed

The term piece part, as used here, is defined as a monolithic integrated circuit, resistor, switch, etc., that is the lowest level of assembly. The next level of assembly is a multi-part assembly that has a defined identity e.g., one that is given a drawing number and, usually, a name. A typical item at this level is a printed wiring assembly (PWA), shop replaceable assembly (SRA), or shop replaceable unit (SRU). The top level is a system. In reality, there is always some aggregate

that is the largest entity reasonably possible to subject to ESS, and there usually are several levels of assembly at which ESS can be considered.

It is more cost effective to do ESS at the lowest level possible and at more than one level. The choices of how many levels and which levels are made on the basis of an engineering evaluation.

The costs associated with a failure usually appear in connection with a single part or interconnection and will increase dramatically with the level of assembly. Consider that:

- At higher levels
  - More assembly work has to be undone and redone when failures occur
  - More material may need to be scrapped
  - More impact on production flow and schedule usually occurs

- At lower levels
  - Corrective action is quicker

In view of the preceding, it is understandable that the tendency is to perform ESS at lower levels of assembly. However, each step in assembly and integration provides additional opportunities for the introduction of flaws. Obviously, ESS at a particular level cannot uncover flaws that are not introduced until the next level. Generally, this dilemma is usually controlled by performing ESS at each major functioning level in the manufacturing process consistent with an assessment of the defect population at each level of assembly.

To resolve these conflicting considerations, screening is usually done at multiple (usually 2 or 3) levels of assembly. ESS at lower levels should focus on precipitating and correcting flaws in piece parts and PWA processing. Most ESS failures at higher levels will reflect flaws introduced later in the manufacturing sequence that are usually correctable without tear-down to a lower level. Table 11.2-5 provides a summary of the risks and results of doing ESS at various levels.

TABLE 11.2-5: RISKS AND RESULTS OF ESS AT VARIOUS LEVELS

ESS CONDITIONS / TRADEOFFS							RISKS / EFFECTS			
Level of Assembly	Power Applied [1]		I/O [2]		Monitored [3]		ESS Cost	Technical		Comments
	Yes	No	Yes	No	Yes	No		Risk	Results	
TEMPERATURE CYCLING										
PWA		X		X		X	Low	Low	Poor	Conduct Pre & Post ESS functional test screen prior to conformal coating
	X			X		X	High	Lower	Better	
	X		X		X		Highest	Lowest	Best	
Unit/Box	X		X		X		Highest	Lowest	Best	If circumstances permit ESS at only one level of assembly implement at unit level
	X			X	X		Lower	Higher	Good	
		X		X		X	Lowest	Highest	Poor	
System	X		X		X		Highest	See Comment		Most effective ESS at system level is short duration random vibration to locate inter-connect defects resulting from system integration.
RANDOM VIBRATION										
PWA	X		X		X		Highest	Low	Good	Random vibration can be effective at PCB level if: 1. Surface Mount Technology is Utilized 2. PWA has large components 3. PWA is multilayer 4. PWA cannot be effectively screened at higher assemblies
	X			X	X		High	High	Fair	
		X		X		X	Low	Highest	Poor	
Unit/Box	X		X		X		Highest	Low	Best	Random vibration most effective at this level of assembly. Intermittent flaws most susceptible to power-on with I/O ESS. Power-on without I/O reasonably effective. Decision requires cost benefit tradeoff.
	X			X	X		Low	Higher	Good	
		X		X		X	Lowest	Highest	Poor	
System	X		X		X		Low	Low	Good	Cost is relatively low because power and I/O normally present due to need for acceptance testing.

NOTES to Table 11.2-5:
1. Power applied - At PWA level of assembly, power on during ESS is not always cost effective.
2. I/O - Equipment fully functional with normal inputs and outputs
3. Monitored - Monitoring key points during screen to assure proper equipment operation

11.2.3.3.2.3 Types and Severities of Stresses

A variety of environmental stresses have been considered for use in ESS over the years. Of these, random vibration and thermal cycling currently are considered to be the most cost effective. Table 11.2-4 identifies some common types of failures and reflects whether random vibration or thermal cycling is the more likely stress to precipitate that particular failure. A failure may be precipitated by one stress, but detected under another.

Traditional ESS, consisting of temperature cycling and random vibration, may not be the most effective. For example, power cycling is effective in precipitating certain types of latent defects; pressure cycling may be desirable for sealed equipment; and acoustic noise may excite microelectronics structures better than structure-borne vibration. Ultimately, ESS environments must be chosen based on the types of flaws that are known or expected to exist.

In the past, fixed-frequency or swept-sine vibration testing was sometimes used. These practices were attributable in part to costs and physical limitations of available test equipment at the time. The shortfalls of fixed frequency and swept-sine vibration in comparison with broadband random vibration were not known at the time. Today, random and quasi-random vibration are used almost exclusively for ESS. It is not difficult to visualize that the complex interactions possible under random vibration can induce a wider variety of relative motions in an assembly.

Burn-in has been defined many ways by different agencies and companies; however, it is most frequently defined as the exposure of powered equipment to either ambient or elevated temperature. Burn-in is not adequate for detecting flaws.

Effective screening requires large, rapid temperature changes and broadband random vibration. Such thermal cycling is used for the detection of assembly flaws that involve installation errors or inadequate chemical or mechanical isolation or bonding. Under rapid thermal cycling, differential thermal expansion takes place without sufficient time for stress relief, and is a major mechanism for precipitating latent defects.

It is important to note that ***thermal cycling and random vibration are synergistic***. For example, thermal cycling following random vibration sometimes leads to detection of vibration-induced failures that were not immediately apparent. In reported cases, vibration stressing had caused a flawed conductor to break, but the loss of continuity only became evident with temperature change. In other cases, a very small flaw may not propagate to the point of detectability during random vibration but may advance to the point of detectability during subsequent thermal cycling.

The concurrent application of the vibration and thermal cycling may be desirable, but is often avoided because it requires more elaborate facilities and makes it more difficult to provide the desired capabilities for functional checking during ESS. Also, concurrent application of random vibration and thermal cycling can make it difficult to determine what caused a defect so that corrective action can be taken. If random vibration and thermal cycling are conducted sequentially, random vibration should be done first.

11.2.3.3.2.4 Failure Detection Measurements During Thermal Cycling and Random Vibration

Two approaches are used to monitor equipment during thermal cycling. In the first approach, limited performance measurements are made prior to and at the end of ESS. These performance measurements may be made on the first and last cycle. Additional measurements may be taken at other cycles, if desired. Each measurement should be made at the hot and cold operating extremes.

In the second approach, equipment operation is continuously monitored during the cold-to-hot transition and during the hot dwell portion of each cycle.

The argument for monitoring equipment during vibration screens is that the resulting movement of a marginal component may show up as an equipment failure only during the stress application. Otherwise, the incipient failure will escape detection, only to show up in an operational environment. Some of the initial work in random vibration screening indicated a 2:1 difference in the efficiency of the screen if the equipment were powered and monitored versus not powered. The technical risks and costs are summarized in Table 11.2-5 at each level of assembly for random vibration screening.

11.2.3.3.2.5 Baseline ESS Profiles

The baseline profiles (Tables 11.2-6 and 11.2-7) represent the combined agreement of the three military services on minimum levels to ensure effectiveness. The profiles are based on experimental and analytical stress screening studies and surveys of screens used in industry. The random vibration baseline profile given in Table 11.2-6 shows a range of recommended minimum acceptable values for input levels, frequencies, axes, duration and monitoring. The thermal cycling baseline profile given in Table 11.2-7 shows a range of recommended values for the temperature extremes, the temperature rate of change and the number of cycles.

TABLE 11.2-6: BASELINE VIBRATION PROFILE

CHARACTERISTIC	LEVEL OF ASSEMBLY		
	PWA [1]	UNIT	SYSTEM
Overall Response Level [2]	6g $_{RMS}$	6g $_{RMS}$	6g $_{RMS}$
Frequency [3]	20 - 2000 Hz	20 - 2000 Hz	20 - 2000 Hz
Axes [4] (Sequentially or Simultaneous)	3	3	3
Duration - Axes Sequentially	10 minutes/axis	10 minutes/axis	10 minutes/axis
- Axes Simultaneously	10 minutes	10 minutes	10 minutes
Product Condition	Unpowered (Powered if purchased as an end item deliverable or as a spare)	Powered, Monitored	Powered, Monitored

NOTES: Pure random vibration or Quasi-random vibration are considered acceptable forms of vibration for the purpose of stress screening. The objective is to achieve a broad-band excitation.

1. When random vibration is applied at the unit level, it may not be cost effective at the PWA level. However, PWAs manufactured as end item deliverables or spares may require screening using random vibration as a stimulus. However, at the system level, when a response survey indicates that the most sensitive PWA is driving the profile in a manner that causes some PWAs to experience a relatively benign screen, that PWA should be screened individually. Each PWA screened separately should have its own profile determined from a vibration response survey.
2. The preferred power spectral density for 6g rms consists of $0.04g^2/Hz$ from 80 to 350 Hz with a 3 dB/octave rolloff from 80 to 20 Hz and a 3 dB/octave rolloff from 350 to 2000 Hz.
3. Vibration input profile for each specific application should be determined by vibration response surveys which identify the correlation between input and structural responses. Higher frequencies are usually significantly attenuated at higher levels of assembly.
4. Single axis or two axis vibration may be acceptable if data shows minimal flaw detection in the other axes.

TABLE 11.2-7: BASELINE THERMAL CYCLE PROFILE

CHARACTERISTIC[1]	LEVEL OF ASSEMBLY		
	PWA [2]	UNIT [3]	SYSTEM
Temperature Range of Hardware	-50°C to +75°C	-40°C to +70°C	-40°C to +60°C
Temperature Rate of Change of Product [4]	15°C/minute to 20°C/minute	10°C/minute to 20°C/minute	10°C/minute to 15°C/minute
Stabilization Criterion	Stabilization has occurred when the temperature of the slowest-responding element in the product being screened is within ±15% of the specified high and low temperature extremes. Large magnetic parts should be avoided when determining that stabilization has occurred. [5]		
Soak Time of Hardware at Temperature Extremes after Stabilization			
- If Unmonitored	5 minutes	5 minutes	5 minutes
- If Monitored	Long enough to perform functional testing		
Number of cycles	20 to 40	12 to 20	12 to 20
Product Condition [6]	Unpowered	Powered, Monitored	Powered, Monitored

NOTES:

1. All temperature parameters pertain to the temperature of the *unit being screened* and not the *chamber air temperature*. The temperature parameters of the unit being screened are usually determined by thermocouples placed at various points on the unit being screened.
2. PWA guidelines apply to individual PWAs and to modules, such as flow-through electronic modules consisting of one or two PWAs bonded to heat exchanger.
3. Unit guidelines apply to electronic boxes and to complex modules consisting of more than one smaller electronic module.
4. Hardware temperature rate of change is limited to capabilities of ESS chambers. The chamber temperature rate of change is optimized to approach the hardware temperature rate of change. This is best accomplished through a series of thermal surveys.
5. It is up to the designer of the screening profile to decide which elements of the hardware (parts, solder joints, PWAs, connectors, etc.) must be subjected to the extreme temperatures in the thermal cycle. The temperature histories of the various elements in the hardware being screened are determined by means of a thermal survey.
6. Power is applied during the low to high temperature excursion and remains on until the temperature has stabilized at the high temperature. Power is turned off on the high to low temperature excursion until stabilization at the low temperature. Power is also turned on and off a minimum of three times at temperature extremes on each cycle.

These minimum acceptable baseline profiles for random vibration and temperature cycling are not recommended stress levels, but are starting points for developing unique optimum profiles for a particular configuration.

The most significant conclusion from ten years of random vibration screening is that the excitation must be tailored to the response experienced by the components of the unit under test. The selection of stress levels must be based on available data and structural design. To avoid potential fatigue or peak level damage due to resonances, some level reduction of the input spectrum may be done at points of severe resonant frequencies (i.e., those which result in amplification of the applied stress level by a factor of 3 dB or more.). These resonances would be obtained from data accumulated during development tests, or by conducting a low-level sine sweep. Where warranted, temporary stiffening of the unit should also be considered to prevent overstressing during the stress screen. The stiffening should be done in a manner which achieves the desired flat response throughout the unit being screened.

The temperature cycling screens also have to be tailored to each specific equipment and are equipment unique. Differences in components, materials and heat dissipation lead to variations in the thermal stresses throughout the item.

11.2.3.3.2.6 Optimizing/Tailoring of ESS

For any given part or production process, there exists a level of ESS stress that is optimal, i.e., maximizes the likelihood of flaw detection without significant degradation of the unit undergoing ESS. ESS tailoring (modification of ESS parameters to fit specific hardware), if not planned and done properly, could be a major expense. Experience with similar hardware can be helpful in setting initial tailoring levels leading to a rough approximation of optimal parameters. However, a true optimization is likely to require an extensive, carefully planned effort.

Recommended tailoring techniques are given in Sections 4 and 5 of the tri-service ESS guidelines for vibration screens and thermal cycling screens, respectively. These are not the only techniques available but are recognized throughout the industry as starting points for an acceptable profile. The selection and use of one or more of these techniques is usually predicated on such things as availability of screening equipment or cost of procurement, architecture of equipment to be tested, type of manufacturing defects expected, and maturity of design and manufacturing processes. Trade-offs are needed because the payoff between reasonably good and optimal ESS parameters may not be commensurate with the costs of finding the optimal profile.

Some specific engineering considerations in determining optimal ESS stress levels and making a sound engineering decision that tends to be on the conservative side (i.e., no overstressing) are as follows:

- Differences in physical characteristics such as thermal inertia, thermal conductivity,

mechanical coupling, and mechanical resonant frequencies assure that differently configured assemblies will respond differently to identical thermal and vibrational inputs.

- Stress profiles should be defined in terms of responses rather than input, especially for vibration. A uniform level of stress may not be achieved throughout the unit, because units are not generally internally homogeneous. The response can be specified and measured at only a few points, so it will still differ locally within differently configured assemblies.

There are various approaches associated with the application of stress screens. Regardless of the approach used, the fundamental objective of ESS remains the same, i.e., to remove latent defects from the product prior to field delivery. The quantitative methods contained in MIL-HDBK-344 and the tri-service ESS guidelines extend this objective by focusing on the defects which remain in the product at delivery and their impact on field reliability.

11.2.4 Production Reliability Acceptance Testing (MIL-HDBK-781)

Reliability acceptance testing is performed on production hardware to determine compliance to specified reliability requirements. MIL-HDBK-781, "Production Reliability Acceptance Testing" contains all the essential procedures and requirements for designing an acceptance test plan for equipment that experiences a distribution of times-to-failure that is exponential. It defines test conditions, procedures and various test plans, and respective accept/reject criteria.

MIL-HDBK-781 has recently been completely revised to include detailed information for test planning and evaluation of data. The latest revision has been restructured to make extensive use of appendices to expand and clarify the various sections of the handbook. It clarifies the definition of mean-time-between-failures (MTBF) and the use of θ_0 (upper test MTBF) and θ_1 (lower test MTBF) , which are test planning parameters, and specifies the use of combined environmental test conditions (temperature, vibration and moisture)* based on the actual mission profile environments encountered during the equipment's useful life.

MIL-HDBK-781 is not intended to be used on a blanket basis, but each requirement should be assessed in terms of the need and mission profile. Appendices are designed so that the procuring activity may reference them along with specific parts of the handbook.

MIL-HDBK-781 covers requirements for preproduction qualification tests as well as production acceptance tests. Qualification tests are normally conducted after growth tests in the development cycle, using initial production hardware to make a production release decision. It should be

* Altitude may be included if the procuring activity determines that it is cost effective, but the cost of test facilities for combining altitude with the other environments would probably be prohibitive.

emphasized that qualification testing, conducted per MIL-HDBK-781, is to demonstrate reliability with statistical confidence, whereas reliability growth testing is performed to improve reliability. Depending on program requirements, funding, and other constraints, preproduction testing may maximize growth testing and minimize statistical testing (resulting in a high MTBF at a low confidence) or may minimize growth and maximize demonstration (resulting in a lower MTBF at a high confidence). Preproduction testing, including both reliability growth and qualification, was discussed in detail in Section 8.

Production reliability acceptance tests per MIL-HDBK-781 are described as "a periodic series of tests to indicate continuing production of acceptable equipment" and are used to indicate individual item compliance to reliability criteria. The tests are intended to simulate in-service evaluation of the delivered item or production lot and to provide verification of the inherent reliability parameters as demonstrated by the preproduction qualification tests. Therefore, an equipment would undergo qualification testing on preproduction hardware.

Once the specified reliability has been demonstrated, and after production begins, the lots produced would undergo reliability acceptance testing, usually at a stress less stringent than the demonstration test level, to indicate continuing fulfillment of reliability requirements.

Production Reliability Acceptance Testing per MIL-HDBK- 781 can be performed based on sampling an equipment from each lot produced as well as on all equipment produced. The test conditions, or stress profile, applied during the test should be measured (preferred) or estimated by the procuring activity and incorporated into the equipment specification. However, when the stress types and levels are not specified by the procuring activity and when measured environmental stresses for the proposed application or a similar application are not available for estimating, then the stress types and levels given in Table 11.2-8, taken from MIL-HDBK-781, should be applied. Table 11.2-8 provides a summary of combined environmental test condition requirements applicable to the following categories of equipment classification:

Category 1: Fixed ground equipment

Category 2: Mobile ground vehicle equipment

Category 3: Shipboard equipment
- sheltered
- unsheltered

Category 4: Equipment for jet aircraft

Category 5: Turbo-prop aircraft and helicopter equipment

Category 6: Air-launched weapons and assembled external stores

Figure 11.2-12, also taken from MIL-HDBK-781, illustrates a typical test cycle that shows the timing of the various conditions. MIL-HDBK-781 describes standard statistical test plans covering:

(1) Fixed length test plans (Test Plans IXC through XVIIC and XIXC through XXIC)

(2) Probability ratio sequential tests (PRST) (Test Plans IC through VIC)

(3) Short run high risk PRST plans (Test Plan VIIC and VIIIC)

(4) All equipment reliability test (Test Plan XVIIIC)

Accept/reject criteria are established on θ_1 and θ_0, where θ_1, the lower test MTBF, is an unacceptable MTBF based on minimum requirements. θ_0 is the upper test MTBF, or the acceptable MTBF. The ratio θ_0/θ_1 is defined as the discrimination ratio. Specifying any two of these three parameters, given the desired producer and consumer decision risks, determines the test plan to be utilized.

Test Plan XVIIIC, shown in Figure 11.2-13, can be used for 100% production reliability acceptance testing. This test plan is to be used when each unit of production (or preproduction equipment if approved by the procuring activity) equipment is to be given a reliability acceptance test. The plan consists of a reject line and a boundary line. The reject and boundary lines are extended as far as necessary to cover the total test time required for a production run. The equation of the reject line is $f_R = 0.72T + 2.50$ where T is cumulative test time in multiples of θ_1 and f is cumulative number of failures. The plotting ordinate is failures and the abscissa is in multiples of θ_1, the lower test MTBF. The boundary line is 5.67 failures below and parallel to the rejection line. Its equation is $f_B = 0.72T - 3.17$.

The test duration for each equipment shall be specified in the test procedure as approved by the procuring activity. The maximum duration may be 50 hours and the minimum 20 hours to the next higher integral number of complete test cycles. If a failure occurs in the last test cycle, the unit shall be repaired and another complete test cycle run to verify repair.

An optional nonstatistical plan can also be used for production reliability acceptance testing. Its purpose is to verify that production workmanship, manufacturing processes, quality control procedures, and the assimilation of production engineering changes do not degrade the reliability, which was found to be acceptable by the reliability qualification test. The test is to be applied to all production items with the item operating (power applied). The required test duration and number of consecutive, failure free, thermal test cycles (minimum of two) which each deliverable

item must exhibit is specified by the procuring activity. The vibration, temperature cycling, and moisture environments together with any others which are deemed necessary may be applied sequentially. The equipment duty cycle and the sequence, duration, levels of the environments, and the vibration option to be used in this test require approval of the procuring activity and are submitted in accordance with the test program requirements.

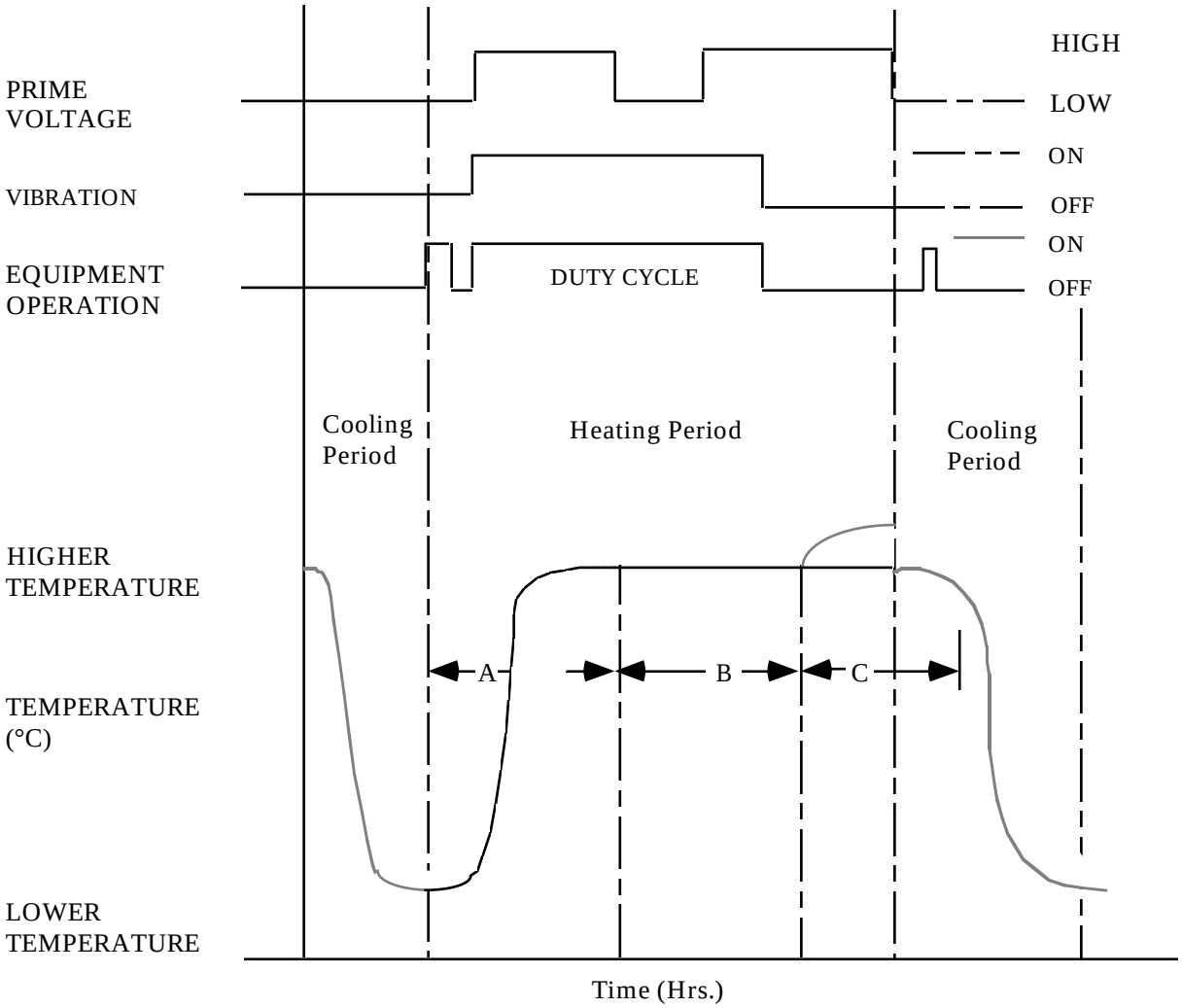
TABLE 11.2-8: TEST CONDITIONS MATRIX (TAKEN FROM MIL-HDBK-781)

Summary of Combined Environmental Test Condition Requirements

	FIXED GROUND	GROUND VEHICLE	SHIPBOARD	
			SHELTERED	UNSHELTERED
ELECTRICAL STRESS				
Input voltage	Nominal +5%-2%	Nominal ± 10%	Nominal ±7%*	Nominal ± 7%*
Voltage cycle	high, nominal and low →	one per test cycle	———	———
VIBRATION STRESS				
Type vibration	sinewave	swept-sine	swept-sine **	swept-sine**
	single frequency	log sweep	continuous	continuous
Amplitude	(See APPENDIX B for	stress levels)		
Frequency range***	20 to 60 Hz	5 to 500 Hz		
Application	20 minimum per equipment	sweep rate 15 minimum once/hr.	(See APPENDIX B ———	———→)
THERMAL STRESS (°C)	A B C ****	LOW HIGH	LOW HIGH	LOW HIGH
Storage temperature	- - -	-54 85	-62 71	-62 71
Operating temperature	20 40 60	-40 TO55	0 TO 50 (CONTROLLED)	-28 65
Rate off change	- - -	5°/min.	5°/min.	5°/min.
Maximum rate of change	- - -	10°/min.	10°/min.	10°/min.
MOISTURE STRESS				
Condensation	none	1/test cycle	See APPENDIX B	1/test cycle
Frost/freeze		1/test cycle		1/test cycle

	AIRCRAFT				AIR-LAUNCHED WEAPONS AND ASSEMBLED EXTERNAL STORES
	FIGHTER	TRANSPORT, BOMBER	HELICOPTER	TURBO-PROP	
ELECTRICAL STRESS					
Input voltage range	nominal ± 10%	± 10%	± 10%	± 10%	± 10%
Voltage cycle	(nominal, high and	low voltage, one cycle	/thermal cycle or per	APPENDIX B)	
VIBRATION STRESS					
Type vibration	random	random	swept-sine log-sweep	swept-sine	swept-sine***
Amplitude	(———	——— SEE	APPENDIX B ——	———	———→)
Frequency range	200 - 2000 Hz	20 - 2000 Hz	5 - 2000 Hz****	10 - 2000 Hz	20 - 2000 Hz
Application	continuous	continuous	sweep rate 15 min. one/hr	continuous (See APPENDIX B)	continuous (See MIL-STD-1670)
THERMAL STRESS (°C)	LOW HIGH	LOW HIGH	LOW HIGH	LOW HIGH	LOW HIGH
Storage temperature (non-oper.)	-54 +71	-54 +71	-54 +71	-54 +71	-65 +71
Operating temperature range	(←———	—— SEE APPENDIX	B ———	———	———→)
Rate of change (min.)	5°/min.	5°/min.	5°/min.	5°/min.	
Duration (nominal)	3 1/2 hours	3 1/2 hours	3 1/2 hours	3 1/2 hours	
MOISTURE STRESS					
Condensation	(1/test cycle ----------	----------	----------	----------	----------)
Frost/freeze	(1/test cycle ----------	----------	----------	----------	----------)

* See MIL-STD-1399
** See MIL-STD-167-1
*** Frequency tolerance ±2 percent or ±0.5 Hz for frequencies below 25 Hz.
**** See 50.1.4 of Appendix B



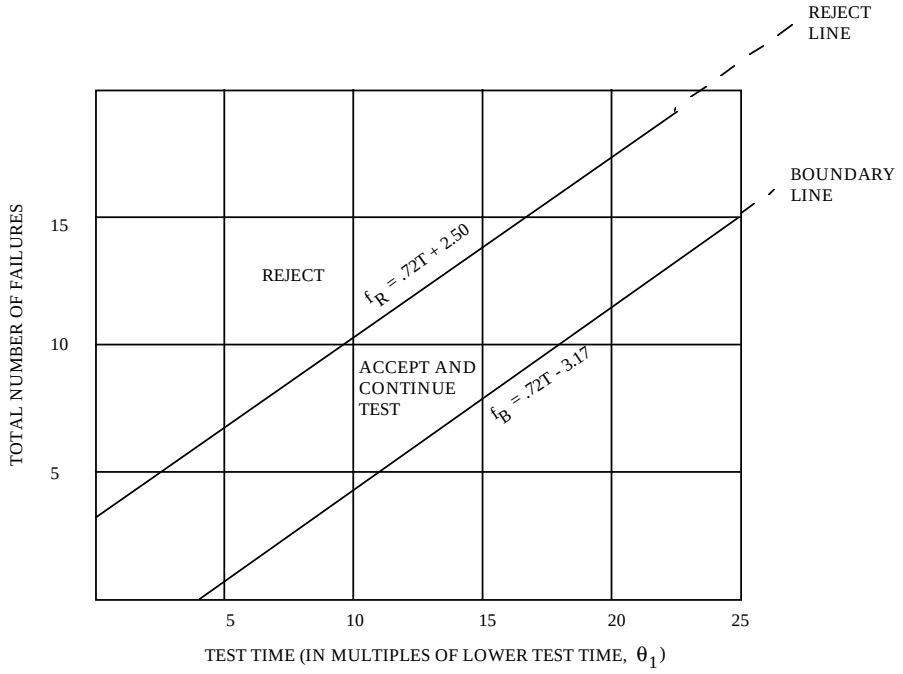
A. Time for chamber to reach stabilization at higher temperature

B. Time of equipment operation at higher temperature

C. Optional Hot Soak and hot start-up checkout

FIGURE 11.2-12: SAMPLE ENVIRONMENTAL TEST CYCLE

SECTION 11: PRODUCTION AND USE (DEPLOYMENT) R&M



Total Test Time*

Number of Failures	Reject (Equal or less)	Boundary Line
0	N/A	4.40
1	N/A	5.79
2	N/A	7.18
3	.70	8.56
4	2.08	9.94
5	3.48	11.34
6	4.86	12.72
7	6.24	14.10
8	7.63	15.49

Total Test Time*

Number of Failures	Reject (Equal or less)	Boundary Line
9	9.02	16.88
10	10.40	18.26
11	11.79	19.65
12	13.18	21.04
13	14.56	22.42
14	etc.	etc.
15	.	.
16	.	.
.	.	.

* Total test time is total unit hours of equipment on time and is expressed in multiples of the lower MTBF. Refer to 4.5.2.4 for minimum test time per equipment.

FIGURE 11.2-13: REJECT-ACCEPT CRITERIA FOR TEST PLAN XVIIIC

It must be emphasized that test criteria, including confidence level or decision risk, should be carefully selected and tailored from these documents to avoid driving cost or schedule without improving reliability. Some general guidelines, taken from MIL-HDBK-781 for planning and implementing production reliability acceptance testing are as follows:

- Production reliability acceptance testing must be operationally realistic, and may be required to provide estimates of demonstrated reliability.

- The statistical test plan must predefine criteria of compliance ("accept") which limit the probability that the item tested, and the lot it represents, may have a true reliability less than the minimum acceptable reliability. These criteria must be tailored for cost and schedule efficiency.

- Production reliability acceptance testing provide a basis for positive and negative financial feedback to the contractor, in lieu of an in-service warranty.

- Production reliability acceptance testing may require expensive test facilities to simulate the item life profile and operational environment; therefore, all equipment production reliability acceptance testing (100% sampling) is not recommended.

- Because it provides a basis for determining contractual compliance, and applies to the items actually delivered to operational forces, production reliability acceptance testing must be independent of the supplier, if at all possible.

- Sampling frequency should be reduced after a production run is well established, however, the protection that it provides for the government (and the motivation it provides for the contractor's quality control program) argues against complete waiver of the production reliability acceptance testing requirement.

Plans for performing production reliability acceptance testing are incorporated into the overall reliability test plan document, and should encompass the following considerations:

(1) Tests to be conducted

(2) Reliability level (i.e., MTBF) to be demonstrated, as well as the associated confidence level, and the relationship between demonstrated MTBF, confidence, test time, etc.

(3) Representative mission/environmental profile

(4) The number of units for test, expected test time, calendar time factors, and scheduling of effort

(5) The kinds of data to be gathered during the test

(6) Definition of failure (relevant, nonrelevant)

(7) Authorized replacement and adjustment actions

(8) Logs/data forms to be maintained that record number of units on test, test time accumulated, failures, corrective actions, statistical decision factors, and accept/reject criteria

11.2.5 Data Collection and Analysis (During Production)

The production reliability test and control program, once implemented in the factory, should continually be challenged relative to the effectiveness of the overall program, as well as that of the individual tests. Production screening and acceptance testing is a dynamic process which must be continually modified in response to experience. Test results and field experience data are monitored to determine the need to modify individual test criteria and conditions to reduce the sampling frequency of acceptance tests and to identify the possibility of applying earlier screen tests where the test costs are less and the potential for cost avoidance is higher. It should be emphasized that the production program, as initially planned, represents a baseline for applying the tests. A production screen test, for example, like any quality inspection, must be adjusted depending on the results of subsequent higher level tests or field performance. However, the extent and nature of any changes should be determined only through careful review and analysis of the subsequent failures.

A data system supported by failure analysis and corrective action is established to maintain visibility over the effectiveness of the production test program as well as all tests including development, qualification, and production. The data system is designed to compile test and failure data and to provide information that would provide a basis to change the test program as necessary to minimize cost and maximize effectiveness. A failure reporting, analysis and corrective action system (FRACAS) is an essential element of the production test program as well as the overall reliability control program. A well designed FRACAS system will provide a uniform mechanism for reporting failures, determining causes and remedies, and making these findings known to the appropriate engineers and designers to enable them to formulate and implement corrective action and, specifically, to ascertain whether or not to design and implement improved inspection, screening and acceptance tests.

Section 8 of the handbook describes failure reporting, analysis, corrective action, and the provisions necessary to assure that failures are accurately reported, thoroughly analyzed, and that corrective actions are taken on a timely basis to reduce or prevent recurrence.

The results of production acceptance test, screening and inspection results, as well as failure reports and analyses from the FRACAS program, are compiled and incorporated into the data

system. Maintaining accurate and up-to-date records through a formal data recording and analysis system is particularly essential in tracking and assessing field reliability performance. Comparative evaluation between predicted reliability estimates and actual field reliability provides criteria for improving production acceptance testing (including the screening and burn-in testing procedures) to assure that the most cost effective test program is developed and applied. This is especially important for new systems where changing performance and reliability characteristics would be expected as a result of design and manufacturing improvements.

A properly designed and operating data system would provide the following information as it pertains to production testing:

(1) Identification of hardware subjected to production tests

(2) Total cumulative operating time for each hardware item including the last operating time interval of failure free operation and acceptance test completion dates

(3) Sampling frequency of reliability acceptance tests

(4) Failure reports of hardware discrepancies including description of failure effects and accumulated operating hours to time of failure

(5) Failure analysis reports of hardware discrepancies including cause and type of failure modes

Also, cumulative plots of screening and burn-in failure events versus time can be prepared and maintained and periodic summary reports submitted to engineering and management activities that provide:

(1) Failure/reject rates by test type and level

(2) Screen test efficiency factors

(3) Responsible failure mechanisms

(4) Recommended or accomplished corrective actions

(5) General product reliability analysis that correlates design predictions with test results and field experience of parts, contamination of surfaces or materials, poor soldering of parts, improper securing of component elements, and bending or deformation of materials

These defects, as mentioned earlier, whether intrinsic to the parts or introduced during fabrication can be further isolated into quality and reliability defects. Quality defects are not time dependent and are readily removed by conventional quality control measures (i.e., inspections and tests). The more efficient the inspection and test the more defects that are removed. However, since no test or inspection is perfect, some defects will escape to later manufacturing stages and then must be removed at a much higher cost or, more likely, pass through to field use and thus result in lower actual operational reliability with higher maintenance cost.

11.2.6 Monitor/Control of Subcontractors and Suppliers

The monitoring of subcontractors is a critical, but often overlooked function of a successful reliability program. End product reliability and its life cycle cost can be adversely affected if a sub-tier vendor or major subcontractor does not fully comply with the applicable reliability program requirements.

The requirements for the monitoring of subcontractors and the monitoring of suppliers often differs due to the nature of the product being furnished and may therefore frequently be defined separately.

11.2.6.1 Major Subcontractor and Manufacturer Monitoring

Development-phase subcontractor monitoring is accomplished by reviewing design data, reliability data, parts selection, non-standard parts requests, failure reports, periodic attendance at design reviews and participation in reliability problem resolution. Production-phase monitoring consists of verifying adherence to the Quality Assurance (QA) standard established between the prime contractor and the subcontractor. It should include the review of process control, production control, personnel qualifications, workmanship and participation in the FRACAS (see section 8.2).

Normally, except for off-the-shelf procurements, the requirements imposed on the manufacturer of a unit/major assembly is as specified in the prime item/system specification.

Supplier monitoring/control requires detailed inspection of the material being furnished, verification of QA procedures, critique of manufacturing processes, periodic inspection to verify adherence to the quality standard, identification of problems, incoming inspection, testing and performance tracking.

Monitoring of Parts Suppliers requires review of vendor performance in addition to the tasks noted.

11.2.6.2 Establishing Vendor Capability and Program Reviews

The most direct method of determining a vendor capability is to review past performance. If this

data is not available or is incomplete, a facility survey should be performed.

Program reviews should be conducted on a continuous basis for the life of the contract. It is essential to develop a free exchange of information and data so that the prime contractor has maximum program visibility into a vendor's process methods and performance. These reviews verify that the manufacturing process is under control, and that: workmanship, personnel certification training, and testing, as defined in the equipment specification and QA manual, is being implemented correctly.

Failure report data from production tests (Burn-in, ESS, PRAT, etc.) received from a vendor or as a result of in-house testing should be reviewed for failure trends and possible corrective action.

11.2.6.3 Supplier Monitoring

Monitoring and verification require that the prime contractor and the selected vendors have a complete and mutual understanding of the standards and quality requirements imposed. A requirements data package should be supplied to each vendor. These data sets then form the foundation for mutual agreement regarding the requirements of a purchase. It is also essential to establish measurement compatibility between the prime contractor's inspection department and the vendor's inspection department should conflicts arise (i.e., a part tests good at vendor final inspection and fails incoming inspection).

Monitoring requirements may vary with the type of procurement (off-the-shelf purchase, etc.). Therefore it is important to assess and plan for the effort that will be required, to define the monitoring requirement associated with the various types of procurement. For example, if component parts are procured from a distributor then the monitoring should consist of verifying that QA and Reliability requirements are developed, that Certificates of Compliance are available, that the Defense Material Administration is monitoring, etc.

11.3 Production Maintainability Control

As was previously indicated for reliability, the inherent design maintainability of an equipment/system can also be degraded during production unless adequate controls are specified and applied to prevent this degradation. This topic is addressed in detail in a companion document MIL-HDBK-470A, "Military Handbook: Designing and Developing Maintainable Products and Systems."

11.4 Reliability and Quality During Shipment and Storage

Electronic components and equipment are subject to change, damage, deterioration and performance degradation during shipment and while in storage. Consequently, the identification

of significant defects, the quantification of the rate at which defects occur, and the analysis of deterioration influenced by shipment and storage environments, dormancy, storage testing, and environmental cycling effects are essential to minimize performance degradation and to assure the designed hardware reliability. Specific inspections and analyses to predict the effects of shipment and storage, to assess the in-storage functional status of component and equipment items, and to control deterioration mechanisms are performed as part of the overall life-cycle reliability program. Included are efforts applicable to:

(1) New Items - determine the environmental conditions needed for proper storage and the effects of shipment, storage and handling on reliability.

(2) Items in Storage - generate storage reliability control techniques covering receipt, storage and prior-to-issue phases of material and equipment items.

The control efforts include identifying components and equipment (and their major or critical characteristics) which deteriorate during shipment and with storage and preparing procedures for in-storage cycling inspection to assure reliability and readiness. The inspection procedures are to identify the number of items for test and the acceptable levels of performance for the parameters under test. Results of these efforts are used to support long term failure rate predictions, design trade-offs, definition of allowable test exposures, retest after storage decisions, packaging, handling, or storage requirements, and refurbishment plans.

11.4.1 Factors Contributing to Reliability Degradation During Shipment & Storage

Defects can be induced during shipment because (1) the packing protection was not adequate for the mode of transportation, (2) the container or other packaging material did not meet specification requirements, or (3) the equipment was roughly handled or improperly loaded.

Electronic components age and deteriorate over long storage periods due to numerous failure mechanisms. In particular, the electrical contacts of relays, switches, and connectors are susceptible to the formation of oxide or contaminant films or to the attraction of particulate matter that adheres to the contact surface, even during normal operation. During active use, the mechanical sliding or wiping action of the contacts is effective in rupturing the films or dislodging the foreign particles in a manner which produces a generally stable, low resistance contact closure. However, after long periods of dormant storage, the contaminant films and/or the diversity of foreign particles may have increased to such an extent that the mechanical wiping forces are insufficient for producing a low resistance contact.

The formation of contaminant films on contact surfaces is dependent on the reactivity of the control material, its history, and the mechanical and chemical properties of the surface regions of the material. Gold is normally used whenever maximum reliability is required, primarily because gold is almost completely free of contaminant oxide films. Even gold, however, is susceptible to the formation of contaminant films by simple condensation of organic vapors and the deposition

of particulate matter. Silver is highly susceptible to the sulfide contaminants that abound in the atmosphere, as are alloys of copper and nickel. Shipping and storage of these systems in paper boxes should be avoided because such boxes contain small amounts of sulfur. Particulate contamination can also lead to corrosive wear of the contact surfaces when the particle is hygroscopic. With this condition, water will be attracted to the contact surface and can lead to deterioration through corrosive solutions or localized galvanic action. The source of such particles can be directly deposited airborne dust or wear debris from previous operations.

Another failure mode which may become significant after long term storage is the deterioration of lubricants used on the bearing surfaces of relays, solenoids, and motors. Lubricants can oxidize and form contamination products. Similarly, lubricants can also attract foreign particles, particularly when exposed to airborne dust, and can lead to lubrication failures and excessive wear.

Over a period of time, many plastics (such as those used in the fabrication of electronic components, i.e., integrated circuits, capacitors, resistors, transistors, etc.) lose plasticizers or other constituents which may evaporate from the plastic, causing it to become brittle, and possibly, to shrink. This can cause seals to leak, insulation to break down under electrical/mechanical stress, and other changes conducive to fatigue and failures. Additionally, plastics may continue to polymerize after manufacture. That is, the structure of the molecules may change without any accompanying change in chemical composition. This will result in change in characteristics and physical properties.

Many materials slowly oxidize, combine with sulfur or other chemicals, or break down chemically over a period of time. These changes may affect electrical resistivity, strength, etc. In addition, many of these materials when exposed to condensed moisture or high humidity conditions may, through a leaching process, lose essential ingredients such as fire retardant additives, thereby causing a hazard to slowly develop. Other materials, such as explosives and propellants, may become unstable over time, posing a safety hazard.

Many component parts and assemblies are sensitive to contaminants and, thus, are sealed during manufacture. These seals will often leak, partly as a result of flexing due to changing temperature and atmospheric pressure, allowing air, moisture or other contaminants to reach the active portions of the component. This leakage can be so slow that the effects may not be discernible for years, but ultimately significant changes can occur.

Finally, the methods/materials of preservation, packaging, and packing (PP&P) used in the storage of components and equipment, i.e., cardboards, plastic bags, polystyrenes, etc., themselves may react with the items stored and cause decomposition and deterioration when left dormant for long durations.

Rough handling during shipment and depot operations, aging, and deterioration mechanisms as

discussed above can, if uncontrolled, lead to a variety of component and equipment failure modes. A summary of some of the failure modes encountered with electronic components during storage is given in Table 11.4-1. Protective measures must be applied to isolate the components from the deteriorative influences in order to eliminate or reduce failure modes such as those listed in Table 11.4-1 and others that can be induced during shipment and storage.

11.4.2 Protection Methods

Proper protection against damage to and deterioration of components and equipment during shipment and storage involves the evaluation of a large number of interactive factors and the use of tradeoff analysis to arrive at a cost effective combination of protective controls. These factors can be grouped into four major control parameters: (1) the level of preservation, packaging and packing (PP&P) applied during the preparation of material items for shipment and storage; (2) the actual storage environment; (3) the need and frequency of in-storage cyclic inspection; and (4) the mode of transportation. These parameters, as depicted in Figure 11.4-1 (circled numbers), must be evaluated and balanced to meet the specific characteristics of the individual equipment and material items. The significance of each of the three parameters is as follows:

(1) Preservation, packaging and packing (PP&P) is the protection provided in the preparation of material items for shipment and long term storage. *Preservation* is the process of treating the corrodible surfaces of a material with an unbroken film of oil, grease, or plastic to exclude moisture. *Packaging* provides physical protection and safeguards the preservative. In general, sealed packaging should be provided for equipment, spare parts, and replacement units shipped and placed in storage. *Packing* is the process of using the proper exterior container to ensure safe transportation and storage.

Various levels of *PP&P* can be applied, ranging from complete protection against direct exposure to all extremes of climatic, terrain, operational, and transportation environments (without protection other than that provided by the PP&P) to protection against damage only under favorable conditions of shipment, handling and storage. A military package as defined per MIL-E-17555, *"Electronic and Electrical Equipment, Accessories, and Provisioned Items (Repair Parts): Packaging of;"* is the degree of preservation and packing which will afford adequate protection against corrosion, deterioration, and physical damage during shipment, handling, indeterminate storage, and worldwide redistribution. A minimum military package is the degree of preservation and packaging which will afford adequate protection against corrosion, deterioration and physical damage during shipment from supply source to the first receiving activity, for immediate use or controlled humidity storage. Many times a minimum military package conforms to the supplier's commercial practice.

TABLE 11.4-1: FAILURE MODES ENCOUNTERED WITH ELECTRONIC COMPONENTS DURING STORAGE

COMPONENT	FAILURE MODES
Batteries	Dry batteries have limited shelf life. They become unusable at low temperatures and deteriorate rapidly at temperatures above 35°C. The output of storage batteries drops as low as 10 percent at very low temperatures.
Capacitors	Moisture permeates solid dielectrics and increases losses which may lead to breakdown. Moisture on plates of an air capacitor changes the capacitance.
Coils	Moisture causes changes in inductance and loss in Q. Moisture swells phenolic forms. Wax coverings soften at high temperatures.
Connectors	Corrosion causes poor electrical contact and seizure of mating members. Moisture causes shorting at the ends.
Relays and Solenoids	Corrosion of metal parts causes malfunctioning. Dust and sand damage the contacts. Fungi grow on coils.
Resistors	The values of composition-type fixed resistors drift, and these resistors are not suitable at temperatures above 85°C. Enameled and cement-coated resistors have small pinholes which bleed moisture, accounting for eventual breakdown. Precision wire-wound fixed resistors fail rapidly when exposed to high humidities and to temperatures at about 125°C.
Semiconductors, Diodes, Transistors, Microcircuits	Plastic encapsulated devices offer poor hermetic seal, resulting in shorts or opens caused by chemical corrosion or moisture.
Motors, Blowers, and Dynamotors	Swelling and rupture of plastic parts and corrosion of metal parts. Moisture absorption and fungus growth on coils. Sealed bearings are subject to failure.
Plugs, Jacks, Dial-Lamp Sockets, etc.	Corrosion and dirt produce high resistance contacts. Plastic insulation absorbs moisture.
Switches	Metal parts corrode and plastic bodies and wafers warp due to moisture absorption.
Transformers	Windings corrode, causing short or open circuiting.

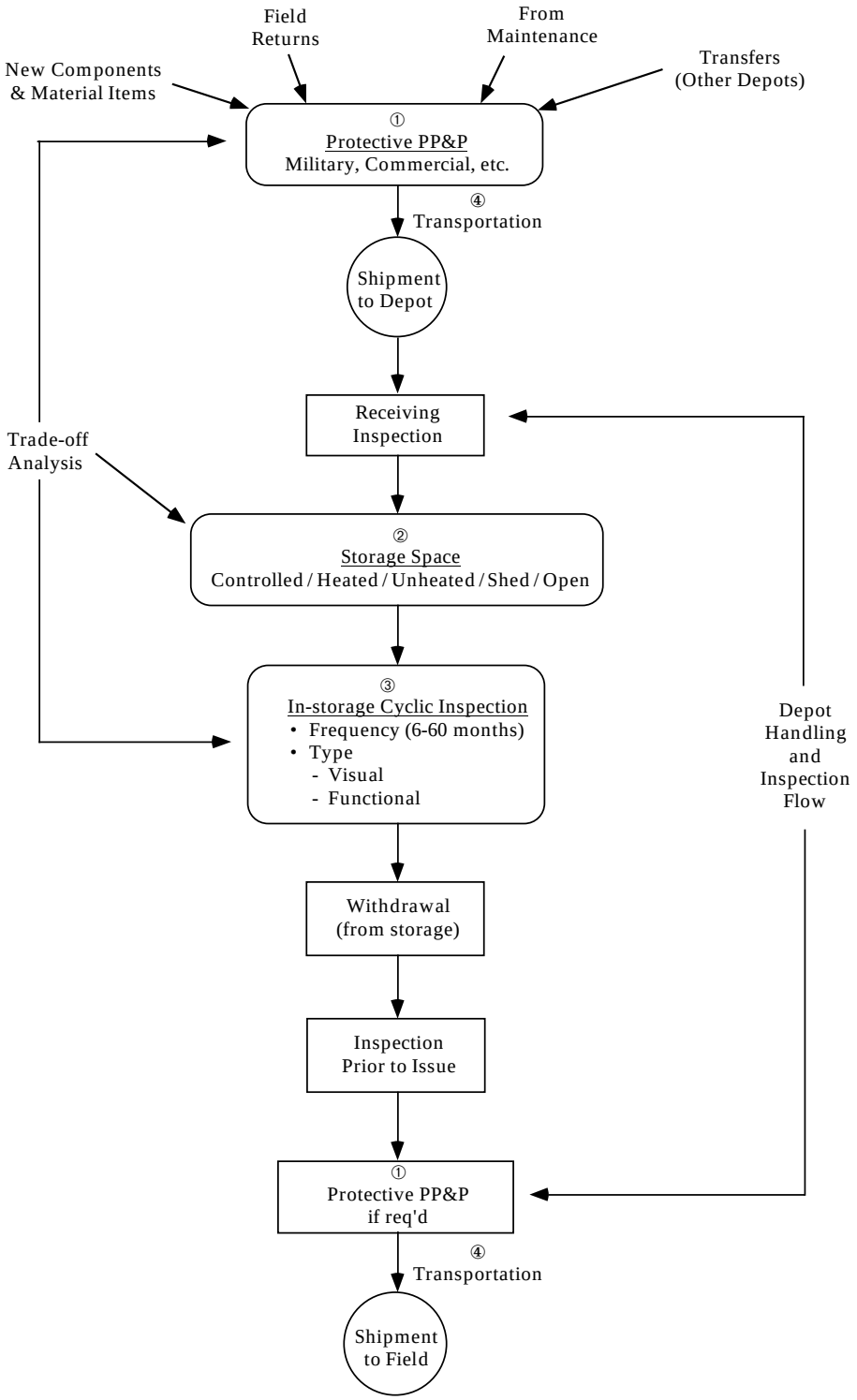


FIGURE 11.4-1: PROTECTIVE CONTROL DURING SHIPMENT AND STORAGE

(2) The storage environment can vary widely in terms of protection afforded. However, whenever possible, electronic hardware should be stored in dry, well ventilated warehouses, where the temperature of the air surrounding the equipment can be regulated so that it does not fall to dewpoint values at night. Storage in controlled temperature/humidity buildings is of course, ideal. If equipment is stored in bins, it is important that it be placed above floor level. The military has several types of storage areas. These include warehouse space with complete temperature and humidity control, warehouse space with no humidity and temperature control, sheds, and open ground areas that are simply designated for storage.

(3) In-storage scheduled cyclic inspection is the key to assuring the actual reliability of components and equipment during storage. In-storage cycling inspections are designed to detect performance degradation, deterioration, and other deficiencies caused by extended periods of storage and improper storage methods. The inspections are to identify those items which require corrective packaging (or further storage control) or condition reclassification to a lesser degree of serviceability. The inspections are performed at intervals derived from shelf life periods and the level of protective packaging and storage afforded the material items. It should be noted that all items when originally placed in storage are ready for issue and that all applicable preservation, packaging and packing (PP&P) requirements have been met. In-storage cycling inspection is part of the depot's overall inspection system (see Figure 11.4-1) that includes inspection of items at receipt as well as prior to issue.

In general, shipment and storage degradation can be controlled in terms of the above-mentioned three parameters. The planning and specification of shipment and storage requirements for new component and equipment items (as well as the reestablishment of requirements for existing items in storage) must take into account economic choices between the various factors within these parameters to arrive at the most cost effective balance that meets reliability and readiness objectives.

(4) The Mode of Transportation greatly influences the level of PP&P needed for an item. The modes of transportation used for military systems are primarily:

- aircraft
- surface ship
- rail
- truck

Each mode is characterized by a unique set of environmental factors. Truck and transport rail, for example, pose a certain temperature and vibration spectrum than do aircraft or surface ships. Exposure times also vary; item shipped by air are exposed to the environmental stresses of transport for a much shorter time than items transported

by rail or surface ship. Items shipped by rail may sit in switching yards or sidings for days under all kinds of climatic conditions. Similarily, an item shipped by air may sit crated on the tarmac under extremes of heat or cold. Items aboard ships may be exposed to highly corrosive salt water spray.

Complicating matters is the fact that most items are not transported from origin to delivery point via a single mode of transportation. An item may, for example, be picked up at its point of origin by truck, driven to a rail loading dock, taken by train to a seaport, sent by ship to another port, downloaded to a truck, and then delivered to its final destination. Such multi-modal transportation imposes a greater variety of environmental stresses. In addition, the handling involved in switching between modes imposes its own set of stresses. The level of PP&P must be sufficient to protect the item against the most severe stresses to which it will be subjected throughout the transportation process.

11.4.3 Shipment and Storage Degradation Control (Storage Serviceability Standards)

Since electronic components and equipment are subject to damage, deterioration and performance degradation if unprotected during shipment and left uncontrolled for long periods of dormant storage, organizations have established programs to control the parameters defined above. The Army, for example, has established the Care of Supplies in Storage (COSIS) program (Ref. [4]). The program assures that material is maintained in a condition to meet supply demands at a minimum cost in funds, manpower, facilities, equipment, and materials. COSIS by definition is "a Department of the Army (DA) program to perform specific tasks to assure that the true condition of material in storage is known, properly recorded, and the material is provided adequate protection to prevent deterioration. The distinction between COSIS-related actions and actions that might otherwise fall into the broad category of care given material in storage is that COSIS concerns itself with the in-storage inspection, minor repair, testing, exercising of material and the preservation, packaging and packing (PP&P) aspects of the efforts."

A major and most significant element within the COSIS program is the Storage Serviceability Standards (SSS) documents controlled by participating Army commodity commands as required by DARCOM-R 702-23, *"Product Assurance - Storage Serviceability Standards (SSSs),"* (Ref. [5]). The SSS documents consolidate and establish the depot quality control and reliability management procedure for inspection, testing, and/or restoration of items in storage. They encompass preservation, packaging, packing (PP&P) requirements, storage environment criteria, as well as inspection requirements during the storage cycle to determine material serviceability and the degree of degradation that has occurred. They are applicable to shelf life items as well as all items that are considered sensitive to shipment and storage deterioration. In the case of shelf life items, specifically those items whose shelf life is considered extendible, the standards are used to determine if the items have retained their original characteristics and are of a quality level which warrants extension of their assigned time period.

Figure 11.4-2 illustrates conceptually the basic technical approach in the preparation of the standards. The figure shows that the storage serviceability standards are formatted into two documents (per Ref. [5]). The first, which is based on Appendix A of Ref. [5], specifies PP&P levels, storage type and those tests, criteria and other provisions that can be coded easily into a computerized format. The second, which is based on Appendix B of Ref. [7], specifies applicable supplementary tests including functional performance, detailed visual and other special tests that cannot be coded easily into a computerized format but are necessary to assess the readiness of the stored items.

The form for the storage serviceability standards (see Figure 11.4-2 and Appendix A of DARCOM-R 702-23) contains in coded format the following data:

Federal Stock Number (FSN) - the federally assigned stock number for the item.

Item Name - provides a brief description of the item.

Quality Defect Code for Inspection (QDC) - defines potential storage-induced defects. The assigned defect codes cover preservation, packaging, marking, and storage as well as material deficiencies. Cyclic inspections are performed to accept or reject material relative to the defects identified by this code. A three-digit code is used, where the first digit identifies the severity of the defect (critical 0, major 1, or minor 2), and the second and third digits (see Table 11.4-2) identify a specific class of defects. For example, the code 113 would indicate a major defect (1) due to (13): container damaged or deteriorated. Complete definitions for quality defect codes applicable to the acceptance/rejection of material items inspected during the various depot inspection and testing phases (i.e., on receipt, audit, scheduled cyclic, special, etc.) are provided in AMCR 702-7 (Ref. [6]).

Inspection Level (IL) - determines the relationship between item lot or batch size and sample size for inspection. The inspection level is used in conjunction with the acceptable quality level (AQL) to form the sampling plan. (The sampling plan provides accept/reject criteria for individual item inspections).

Acceptable Quality Level (AQL) - the maximum percent defective (or the maximum number of defects per hundred units) that for purposes of sampling inspection can be considered satisfactory.

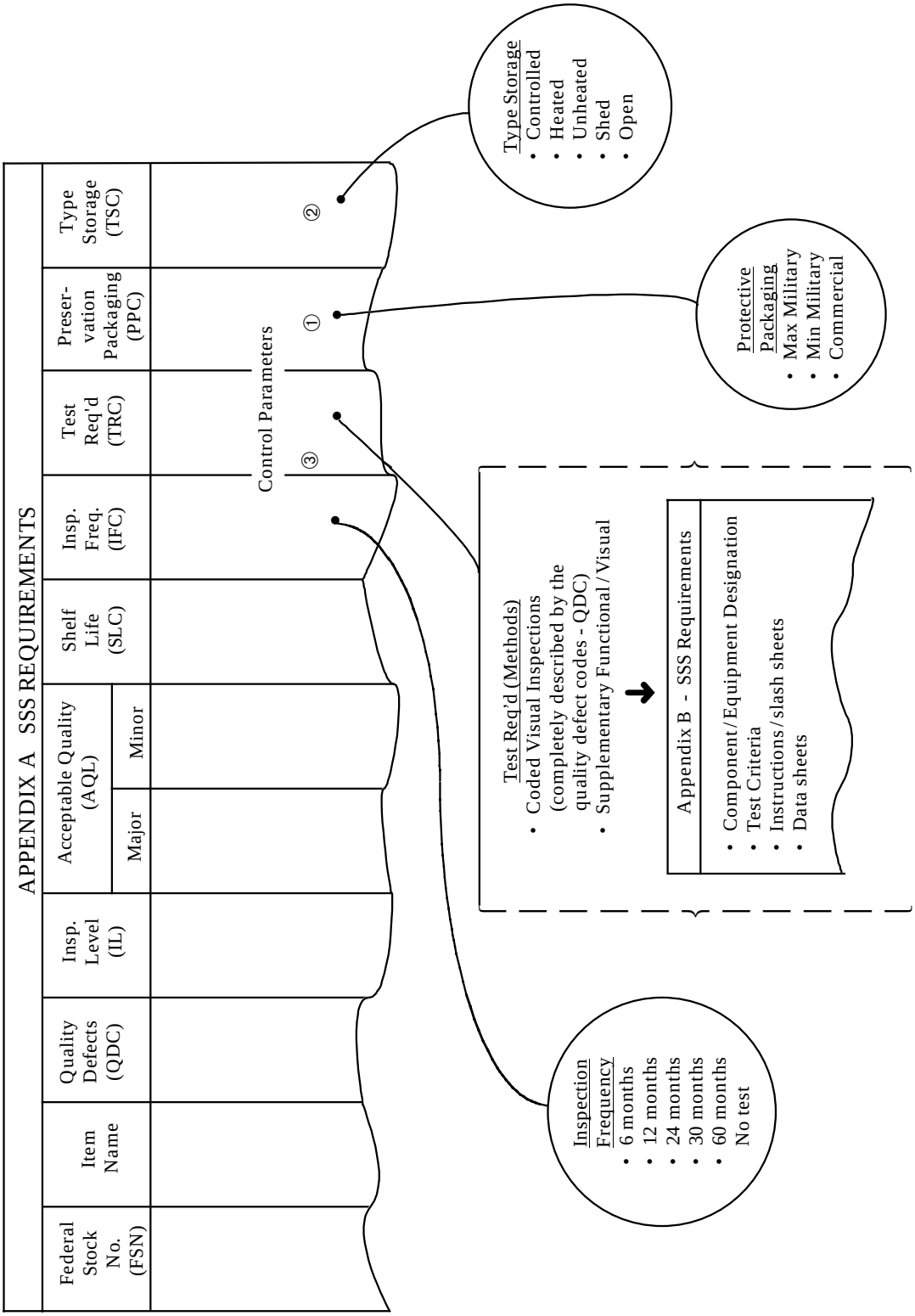


FIGURE 11.4-2: TECHNICAL APPROACH TO STORAGE SERVICEABILITY STANDARDS (SSS)

TABLE 11.4-2: STORAGE-INDUCED QUALITY DEFECTS

Category	Second & Third Digit (QDC)
Preservation Inadequate	02
Container Damaged or Deteriorated	13
Containers, Boxes, Crates, or Pallets Damaged or Deteriorated	23
Markings Illegible	33
Loose or Frozen Parts (out of adjustment)	40
Damaged Parts (cracked, chipped, torn)	41
Leakage (liquid)	45
Bonding Deterioration (soldering, welding, etc.)	48
Contamination (dirt, sludge, moisture, foreign matter)	50
Excessive Moisture (fungus, mildew, rot)	51
Shelf-life Data Exceeded	55
Failed Test Requirements (failed supplementary tests functional/visual)	62
Improper Storage Space	86
Corrosion, Stage 1 (or more)	90

Shelf Life (SLC) - describes deterioration characteristics versus time. Shelf life periods for deteriorative material range from 1 month to 60 months. The condition of a shelf-life item is evaluated during cyclic inspection in terms of time remaining and downgraded if necessary.

Inspection Frequency (IFC) - defines the elapsed time between cyclic inspections. Inspection periods range from 6 months to 60 months.

Test Required (TRC) - describes the method by which an item is to be inspected or tested.

Preservation Packaging (PPC) - describes the preferred level and/or most cost effective level of protection for each item. After an item has been inspected and accepted, the packaging/preservation is to be restored to its pre-inspection level. Further, the date of repackaging as well as the date of original packaging is stamped on the package.

Type Storage (TSC) - indicates the preferred or most cost effective storage condition.

In order to prepare standards for new or existing material items, criteria for specifying cost effective tests and control provisions are first established. The criteria (and the subsequent standards) should provide for the inspections to be performed frequently enough to detect

potential problems but not so often as to dilute the total depot inspection effort and compromise other items in storage which may be more critical and require higher inspection frequencies. To be effective, the criteria must take into account:

(1) Material deterioration

(2) Application risk and criticality

(3) Cost

(4) Material complexity

(5) Preservation/packing and packaging (PP&P)

(6) Storage environment

The Army has developed general criteria and a material weighting factor technique as part of a complete standard preparation process that takes into account these factors (Ref. [7]). The process, which is illustrated in Figure 11.4-3, focuses on the three major control parameters: (1) protective packaging level, (2) storage type, and (3) cyclic inspection (frequency and method). The process involves first defining the level of packaging and storage (preferred) from a review of material deterioration properties and then determining inspection frequency by evaluating deterioration, application risk, criticality and other factors in light of the defined packaging and storage environment levels. It is an iterative process that involves tradeoff analysis to define an optimum set of requirements. It emphasizes and uses to the maximum extent the visual coded inspection criteria, i.e., QDC, to detect material failure and/or defects due to corrosion, erosion, and other deficiencies resulting from improper storage methods, extended periods of storage, and the inherent deterioration characteristics of the material item. The technique is sufficiently flexible to make allowances for available storage facilities if they differ from the preferred through the adjustment of inspection frequency.

In the initial preparation of the standards, the type and level of storage space and packaging methods are considered as fixed parameters (although iterative) where the preferred levels are defined based on material deterioration properties. Therefore, the element which provides the overall stimulus for the control and assurance of the readiness of stored components and equipment is the type and frequency of inspection. A ranking is assigned to each item that accounts for material deterioration and the other factors depicted in Figure 11.4-3 and is used as the basis to determine first the need for inspection and then, if needed, the frequency and type of inspection.

To effectively manage the depot cyclic inspection program, priorities are established as indicated in Figure 11.4-3. Items classified as definite shelf-life are given priority and subjected to cyclic inspection. Other indefinite shelf-life items that are considered particularly sensitive to

deterioration are also subject to cyclic inspection. Definite shelf-life items are those possessing intrinsic deterioration characteristics that cannot be eliminated (or minimized) by storage and packaging controls. They are further classified into nonextendible (Type I) and extendible (Type II) materials. Indefinite shelf-life items, on the other hand, include items that do not deteriorate with storage time, as well as items that are sensitive to deterioration as a result of induced external failure mechanisms. The relationship between these types of material item classification and their relative deterioration level is illustrated in Figure 11.4-3. Figure 11.4-4 shows the nonextendible life characteristic of Type I material, the extendible shelf-life characteristic of Type II material, and the relative indefinite shelf-life characteristic of all other stored material.

Figure 11.4-5 presents a matrix that can be used to determine inspection frequency (IFC) and to optimize in-storage inspection coverage. The matrix includes:

(1) The most deteriorative items to the least deteriorative in terms of a total ranking factor that accounts for deterioration, complexity, cost, accessibility and criticality

(2) All combinations of depot storage and packaging conditions ranging from the most protective (containerized package and a controlled humidity environment) to the least protective (commercial package and an open area)

Application of the matrix to a given material item involves assigning appropriate values to each of the weight factors depicted in Figure 11.4-5 in order to arrive at a total ranking. This ranking represents a rough measure of the overall deterioration/cost sensitivity of the item to the storage environment. The ranking is then entered in the proper weight column of the matrix to determine inspection frequency for any desired combination of packaging and depot storage protection level.

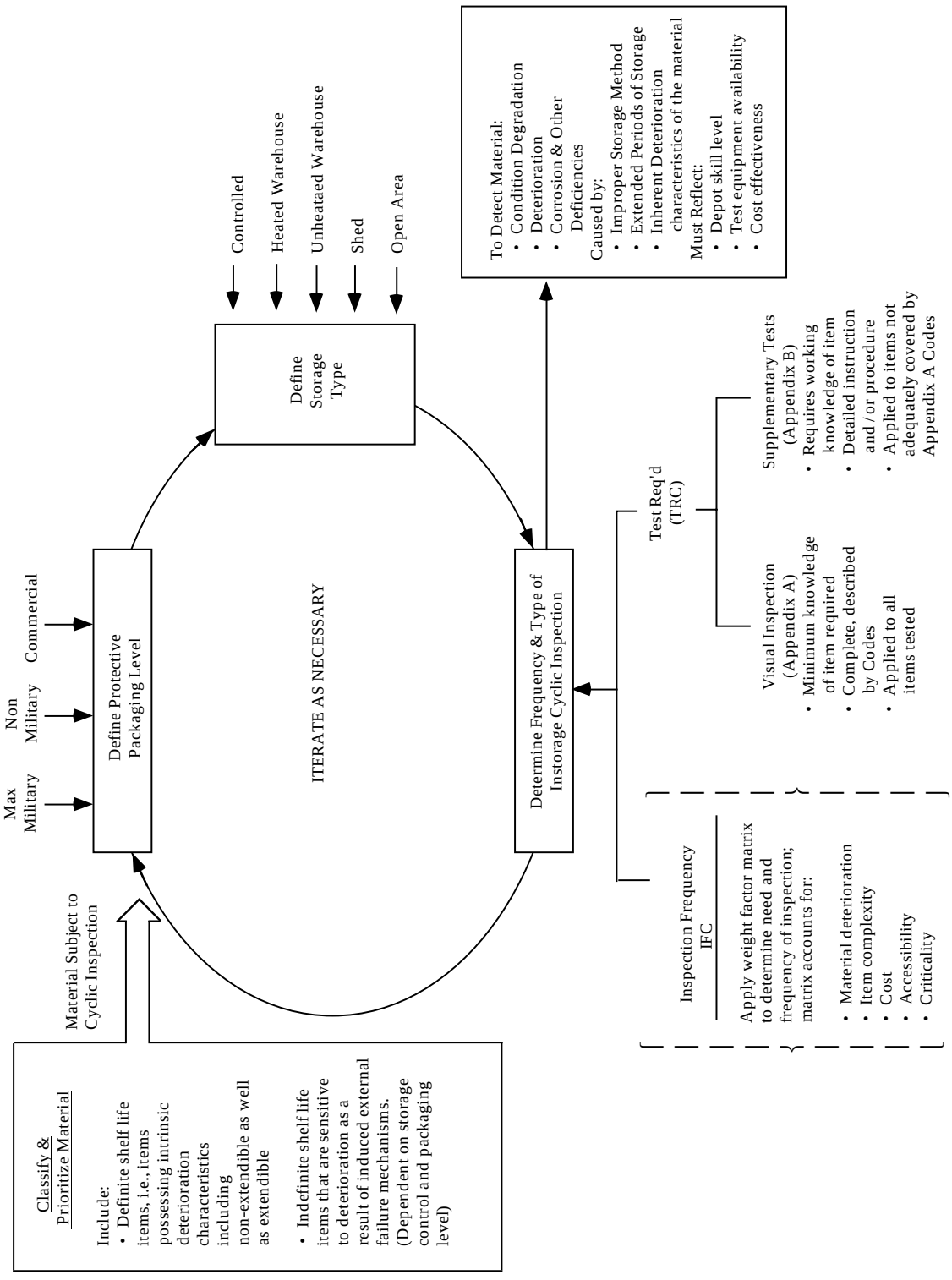


FIGURE 11.4-3: STORAGE SERVICEABILITY STANDARD PREPARATION PROCESS

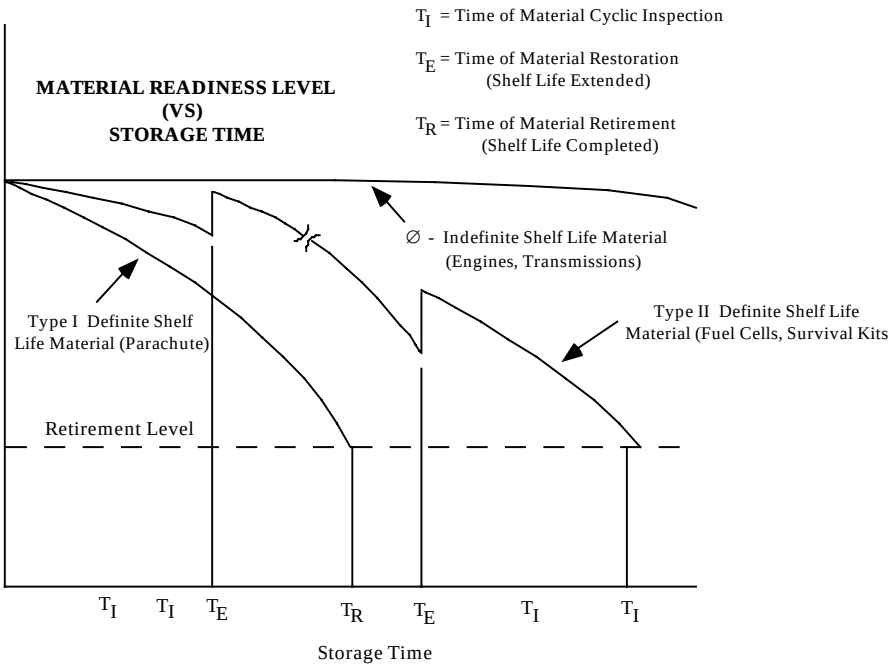


FIGURE 11.4-4: DETERIORATION CLASSIFICATION OF MATERIAL

STORAGE PROTECTION		Test Frequency Code					
Storage Environment	Packaging Level						
Controlled Humidity	Containerized	6	6	6	5	3	2
Heated	Containerized	6	6	6	5	3	2
Unheated	Containerized	6	6	6	5	3	2
Shed	Containerized	6	6	6	5	3	2
etc.	etc.	etc.	etc.	etc.	etc.	etc.	etc.
Open	Commercial	6	3	2	2	1	1
Material Weight Factor		0-1	2-3	4-5	6-7	8-10	11-12

TEST FREQUENCY	CODE
6 months	1
12 months	2
24 months	3
30 months	4
60 months	5
No test	6

	WEIGHT FACTOR
DETERIORATION	
LOW	0
MODERATE	1
HIGH	2
COMPLEXITY	
LOW	0
HIGH	1
ITEM COST	
LOW	0
MEDIUM	1
HIGH	2
ACCESSIBILITY	
(IMPACT ON SYSTEM REPAIR TIME)	
- NO MAJOR EFFECT, SIMPLE SUBSTITUTION OF REPLACEABLE ITEM (I.E., EASILY ACCESSIBLE)	0
- NOT READILY ACCESSIBLE, REPAIR TIME INVOLVED, REQUIRES SOME SYSTEM TEARDOWN	1
- NOT ACCESSIBLE, REPAIR TIME IS SUBSTANTIAL, REQUIRES MAJOR SYSTEM TEARDOWN	2
CRITICALITY	
LOW	0
MEDIUM	2
HIGH	5

FIGURE 11.4-5: INSPECTION FREQUENCY MATRIX

For new items, the matrix allows broad tradeoffs to be made to arrive at the optimum balance of packaging, storage, and inspection requirements. Also, the combining of deterioration with cost and the other weight factors via the matrix approach allows the specification of cost effective inspection periods. This cost effectiveness is illustrated by considering two items one of which exhibits low deterioration properties but the cost and other factors are high, and the other which exhibits high deterioration properties but the total of the other factors is low. A relatively low cost or nominal test inspection frequency may be computed for both items that reflects an effective balance of all factors; whereas, if only deterioration was considered in computing the test periods, over-inspection (excessive cost) of the high deterioration item and under-inspection (low readiness assurance) of the low deterioration items would most likely result. Of course, for those items where all factors including deterioration and cost are high, frequent inspection would be required to ensure the readiness of material and for those items where deterioration and the other factors are low, less frequent inspections would be required.

The matrix approach also provides flexibility for regulating the number and type of items subjected to cyclic inspections by adjustment of the weight assigned to the factors that relate the material to the storage environment.

As previously indicated, an inspection time period is originally set based upon preferred storage environment and packaging methods specified in the TSC and PPC columns of Figure 11.4-2. However, many times an item will be stored and packaged at a different level. In that case an adjustment is made to its inspection time periods to maintain the same state of readiness based on the data provided in the inspection frequency matrix.

11.4.3.1 Application of Cyclic Inspection During Storage to Assure Reliability and Material Readiness

Critical to the control of reliability during storage is the proper application of cyclic inspections and tests. The purpose of in-storage cyclic inspection is to assess component/equipment reliability and readiness for use, to detect deterioration while in storage, and to furnish data for any necessary condition reclassification action. A knowledge of the component or equipment item, particularly its deterioration properties and risk attributes, is necessary to plan and specify optimum in-storage cyclic inspection requirements. The inspections must be practical and maintain an overall cost effective posture that reflects readily available depot test equipment and skills.

In-storage cyclic inspection generally includes two basic types as indicated in the previous subsection. The first type is based on subjective visual inspections where material acceptance is completely described by codes covering quality defects (and included in the QDC column of the Storage Serviceability Standard). A minimum knowledge of the items is required to specify the criteria and perform the inspections. These coded requirements apply to all items tested. Figure 11.4-6 illustrates some of the quality defect codes and shows that the assigned codes cover preservation, packing, marking and storage, as well as material deficiencies. The figure indicates that there are basically three levels of inspection corresponding to (1) the outer package or container, (2) the inner packing, and (3) the item itself. If a defect is not considered critical, major, or minor at the time of inspection but (due to inspector experience) is expected to become critical, major or minor prior to the next cyclic inspection, it is identified as such, considered as a cause for rejection, and counted relative to the item's sampling plan criteria. Defects of a trivial nature are not considered as cause for rejection of a lot, unless some reduction in usability or function of items is expected prior to the next scheduled inspection. For example, nicks, dents, or scratches that do not break coatings or paint films are considered trivial deficiencies.

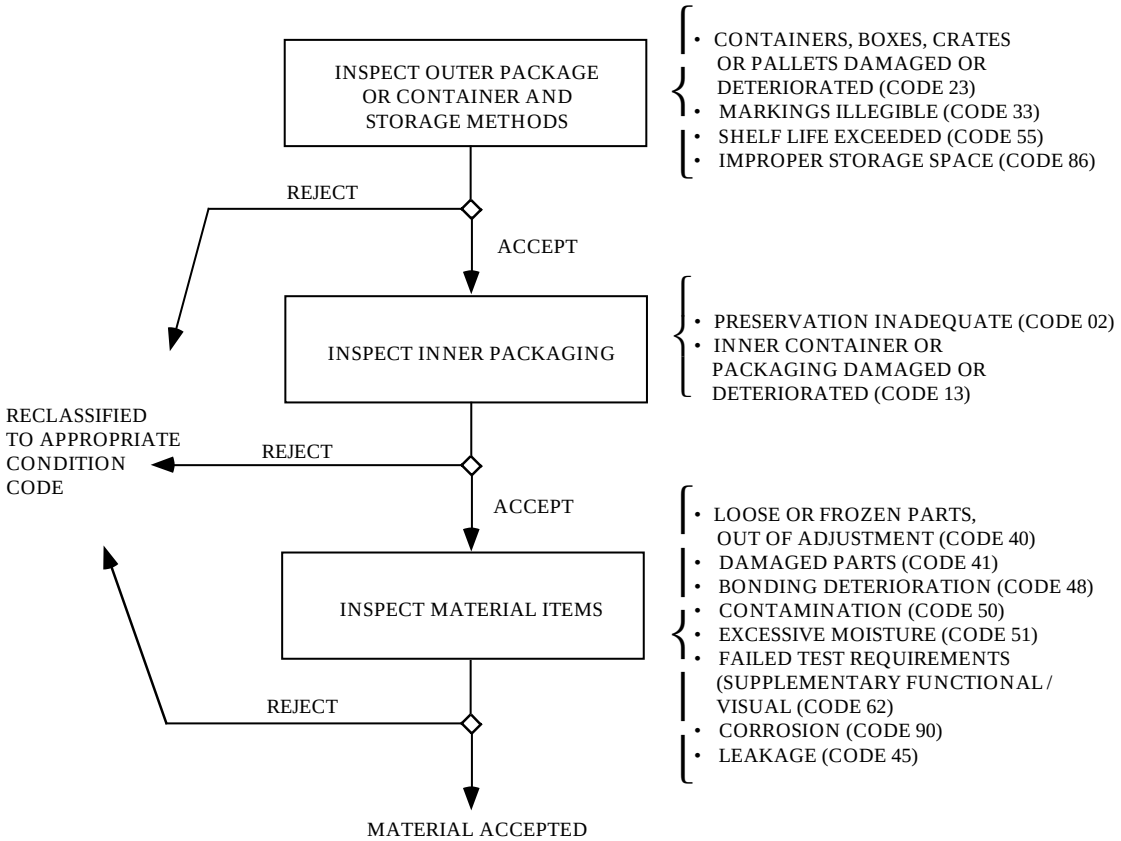


FIGURE 11.4-6: CODED QUALITY INSPECTION LEVELS

The second type of in-storage inspection involves supplementary requirements that are applied to items that cannot adequately be inspected by the visual coded requirements. They generally include functional tests (derived from technical manuals) and/or special, more-detailed visual inspections. Special test and/or inspection procedures complete with acceptance criteria are prepared for these items and included in Appendix B to the SSS. Emphasis is placed on defining viable test or checkout procedures that can be applied simply and quickly to the stored material items to assure that they perform satisfactorily with only a minimal level of evaluation, support, and guidance. These supplementary tests can be applicable to parts, material, equipment, or complete systems, including shelf-life items as well as other items that are storage sensitive.

The supplementary tests are not intended to represent a complete and detailed inspection or checkout of the item to determine compliance to specified requirements. The tests are designed to verify operability and are to be based on a "go/no-go" concept, fully utilizing end item functions to indicate functional readiness for service and issuance.

The functional tests are designed such that they do not require external and specialized test equipment except common and readily available equipment found at the depots and other installations (power supplies, volt-ohmmeters, etc.). The functional tests in general involve first checking the operational mode of all indicators such as dial lamps, power lights, meters, and fault lights as applicable and then applying a simple procedure that exercises some or all of its functions to verify operational status. Many times the equipment can be tested as part of a system. For example, two radio (receiver/transmitter) sets could be tested as a system pair by positioning the sets a certain distance apart (e.g., 25 feet). One is placed in the receive mode and the other in the transmit mode, and all associated hardware and interconnecting cables are attached. An audio (spoken word) input is applied to the set in the transmitting mode, and the set in the receive mode is checked for reception. The process is repeated with the transmitter/receive modes reversed.

The functional test procedures for a given equipment item can be derived from a review of the equipment's maintenance and/or operating manuals. These manuals describe the operational sequence, the turn-on and shut-down procedure, and the equipment operational test and checkout procedure necessary for complete verification of equipment operational status. Consequently, they provide a sound basis for deriving a simplified and cost effective functional test that is suitable for assessing reliability during storage.

11.4.4 Data Collection and Analysis (During Storage)

The shipment/storage test and control program, like the production test program, must be continually challenged relative to the effectiveness of the overall program as well as the individual tests. In-storage cyclic inspection must also be considered as a dynamic test where the test methods, frequencies, and criteria are adjusted to reflect actual depot and field experience. In-storage data (reject rate, quality discrepancy reports, causal data, etc.) generated during the implementation of the cyclic inspections should be compiled, reduced, thoroughly analyzed, and fed back to item management and engineering activities in a form that will provide a basis to:

(1) Determine the effectiveness of the shipment/storage degradation control program to meet reliability and readiness objectives

(2) Eliminate the causes for deficiencies

(3) Revise item inspection or protective packaging and storage level requirements, if necessary

11.5 Operational R&M Assessment and Improvement

Electronic systems are also subject to damage and performance degradation during operation and maintenance. Consequently, operational systems are continually assessed to ensure that they are performing in accordance with expectation and to identify areas where improvements can be incorporated to minimize degradation, improve R&M, and reduce life cycle costs. This time period is most significant because it is here that the true cost effectiveness of the system and its logistic support are demonstrated and historical R&M data are gathered and recorded for use on future products. The effort includes:

(1) Assessing R&M performance from an analysis of operation/failure data, identifying operation/maintenance degradation factors, and comparing actual R&M with that predicted and demonstrated during acquisition

(2) Identifying systems, equipment and other hardware items that exhibit poor reliability, require extensive maintenance and are prime candidates for cost effective improvements

(3) Evaluating the impact on R&M of system changes and corrective action implemented in response to operational failures

11.5.1 Factors Contributing to R&M Degradation During Field Operation

Degradation in reliability can occur as a result of wearout, with aging as the dominant failure mechanism. Defects can also be induced into a system during field operation and maintenance. Operators will often stress a system beyond its design limit either to meet a current operational need or constraint or inadvertently through neglect, unfamiliarity with the equipment, or carelessness. Situations occur in which a military system may be called upon to operate beyond its design capabilities because of an unusual mission requirement. These situations can cause degradation in inherent R&M parameters. Operational abuses due to rough handling, extended duty cycles, or neglected maintenance can contribute materially to R&M degradation during field operation. The degradation is usually the result of the interaction of man, machine and environment. The translation of the factors which influence operational R&M degradation into corrective procedures requires a complete analysis of functions performed by man and machine plus environmental and/or other stress conditions which degrade operator and/or system performance.

Degradation in inherent R&M can also occur as a result of poor maintenance practices. Studies have shown that excessive handling brought about by frequent preventive maintenance or poorly executed corrective maintenance (e.g., installation errors) have resulted in defects introduced into the system, with resultant degradation of R&M. Some examples of defects resulting from field maintenance, depot overhaul, or reconditioning are due to:

(1) Foreign objects left in an assembly

(2) Bolts not tightened sufficiently or overtightened

(3) Dirt injection

(4) Parts replaced improperly

(5) Improper lubricant installed

Also, during unscheduled maintenance, good parts are replaced in an effort to locate the faulty parts. In many cases, the good parts are written up as defective instead of being reinstalled. These parts often are returned to the depot for repair or discarded, resulting in a reported field failure rate that is higher than is actually occurring.

Several trends in system design have reduced the need to perform adjustments or make continual measurements to verify peak performance. Extensive replacement of analog with digital circuitry, inclusion of more built-in test equipment, and use of fault-tolerant circuitry are indicative of these trends. These factors, along with greater awareness of the cost of maintenance, have brought changes for ease of maintenance whose by-product has increased

system R&M. In spite of these trends, the maintenance technician remains a primary cause of R&M degradation. The effects of poorly trained, poorly supported or poorly motivated maintenance technicians on R&M degradation require careful assessment and quantification.

The operation and maintenance induced defects are factors that must be carefully considered and taken into account in the assessment and control of operational R&M. In general, the environmental factors considered in prediction techniques account for the added stress provided by operation within that environment. However, the environmental stresses imposed during field maintenance may be other than what was anticipated during the original prediction. For instance, a subassembly removed for repair in a desert area may be placed in direct sunlight while awaiting transfer. Component temperatures may exceed those experienced during normal operation for an extended period, thus reducing their life expectancy. Mechanical stresses imposed on components during removal, repair, and reinsertion may exceed that designed for a given environment. Therefore, field and depot requirements and procedures must include criteria for controlling the reliability and quality of the repair/overhaul action to minimize potential maintenance induced defects in order to achieve an actual field R&M that approaches that predicted and demonstrated during acquisition.

11.5.2 Maintenance Degradation Control (During Depot Storage)

Depot maintenance activities include complete overhauling, partial rebuilding, product improvement and retrofit, calibration, and the performance of highly complex repair actions. In addition, the depot normally stores and maintains the supply inventory. Physically, depots are specialized fixed installations that contain complex and bulky production and test equipment, and large quantities of spares under environmental storage control. Depot facilities maintain high volume potential and use assembly line techniques with relatively unskilled specialists in key areas such as condition evaluation, fault diagnosis, and quality control and inspection.

Since the R&M of hardware items can be materially degraded during maintenance and depot operations, engineering plans and analyses are performed and R&M controls implemented to assure performance and to eliminate defects due to workmanship and the various other factors that would, if uncontrolled, lead to poor quality and R&M degradation.

Control efforts for a given hardware item start with the preparation of a Maintenance Plan during development as part of logistic support analysis (LSA); they continue into the operational and maintenance phase with the establishment of specific criteria and special maintenance and restoration procedures which must be followed to avoid R&M degradation and to retain the inherent R&M of the item. Possible deviations from the Maintenance Plan are described and related to their potential effect on operational R&M. Specifications are prepared and incorporated into a maintenance/depot requirement document including provisions covering:

(1) Life cycle reconditioning performance/quality parameters and acceptance criteria

(2) Types and kinds of material approved for use during overhaul, repair, and reconditioning

(3) Acceptable workmanship standards and techniques

(4) Quality and reliability assurance inspection, tests, analysis methods, and controls

The intent of the maintenance requirement document is to ensure that quality and R&M measures reflect adequate, viable, and practical acceptance criteria and procedures that can be implemented most cost effectively by depot personnel during the repair, overhaul, or reconditioning of the hardware items.

Some of the areas that are evaluated, controlled and reflected into the maintenance documentation from a reliability and quality standpoint are listed in Table 11.5-1. These include reviewing the technical accuracy and adequacy of instructions covering equipment checkout, calibration, alignment, and scheduled removal and replacement. In addition, all disassembly, cleaning, inspection, testing, repair, replacement, re-assembly, troubleshooting, preventive maintenance checks and services, and maintenance processes and procedures are evaluated.

Criteria are also established that recognize the fact that hardware in field use (as well as during storage) deteriorates due to age, environment, and storage conditions. When deterioration begins to take effect, the quality level of the material will decline below that which was initially specified during procurement. Although the effectiveness and adequacy of the reconditioning operations and controls will minimize the decline, the resultant quality level of the reconditioned material will usually be lower than that initially specified. The depot requirements include maintenance quality level requirements that reflect:

(1) Minimum deterioration, which is lower than the initially specified value

(2) Criteria that indicate the quality limits beyond which repair is not economically achievable

(3) Acceptance criteria for reconditioning cycles(s) at predetermined storage and use milestones

TABLE 11.5-1: DEPOT MAINTENANCE REQUIREMENT AREAS

Inspection and Test Equipment - Test equipment used to determine performance of depot maintenance specifications and requirements

Material Quality - Quality level of parts and material used for replacement, repair or modification

Pre-shop Analysis - Extent of overhaul required. Included in the analysis would be procedural instructions as well as a detailed checklist to aid in the evaluation of the items for determining extent of cleaning, repair, modification or replacement

In-Process Inspection - In-process inspection requirements, including procedural as well as accept/reject criteria associated with each overhaul operation such as disassembly, cleaning, repair, replacement and modification, as applicable

Diagnostic and Automated Test Equipment - Diagnostic and automated test equipment (such as NDT, magnetic particle, dye penetration, etc.) used to determine the adequacy of repair, overhaul or reconditioning

Repair - Total sequential, step-by-step instructions and specifications used for repair, replacement, reclamation, rework or adjustment for hardware items

Assembly/Disassembly - Total step-by-step instructions used to assemble/disassemble the hardware item

Calibration - Level and method of calibration for all equipment and instrumentation

Final Performance Check - Techniques and methods to assure total satisfactory performance of the hardware item in accordance with the established criteria

In addition, a process analysis similar to that described in Sections 11.2 and 11.3 to determine and control R&M degradation introduced by manufacturing can also be applied to determine and control degradation introduced by the reconditioning and overhaul process. This analysis would identify processing and inspection steps that can be improved to reduce R&M degradation and determine the need to incorporate controlled screening and burn-in tests as described in Section 11.2.

11.5.3 Maintenance Documentation Requirements

An important factor in controlling R&M degradation during deployment is the availability of adequate maintenance documentation for the equipment/system. System maintenance documentation includes the written, graphical, and pictorial data which should be supplied with the system for use by operators and maintenance personnel to accomplish both the routine preventive maintenance tasks and the corrective repair procedures identified in the Maintenance Plan for the system. This documentation should reflect the maintenance concept and repair policies established for the system. In general, system operation and maintenance documentation should be a completely integrated package providing clear-cut direction leading from failure detection to fault isolation and repair procedures and should be presented in a format and style designed for ready access and updating as changes are introduced.

Four types of data represent the minimum package which should be provided with an operating system if it is to be successfully operated and maintained in accordance with the Maintenance Plan. These working documents should be instructional and factual. The four categories of maintenance documentation required to successfully implement the Maintenance Plan are described as follows:

(1) Functional Description and Operating Instructions for Each System - Data in this category includes: a description of the capabilities and limitations of the installed system; a technical description of system operation, including its operating modes and alternate modes; step-by-step turn-on and manual operating procedures; “confidence” checks normally employed to verify that equipment is performing satisfactorily.

(2) Equipment and Installation Description - Data in this category must provide an accurate, up-to-date description of the hardware as it is installed in the weapons system. Minimally, it should consist of: A complete set of functional flow or logic diagrams; a complete set of schematic diagrams for electrical layout, electronics, hydraulics, pneumatics, etc.; parts data containing reference information in sufficient detail to permit reordering or fabrication of the individual parts within the system; and the necessary instructions for installing and checking out installed/retrofitted equipment.

(3) Maintenance Aids (Troubleshooting) - This category presents the specific data required by the technician for localizing a fault to a replaceable item and for checking out the system after repair. Included are:

    (a) Methods for system-level fault isolation when the system is “up” but operating in a degraded mode; use and interpretation of system readiness test results

(b) Method of system level fault isolation when the system is totally down; use and interpretation of fault isolation tests and monitoring of console displays

(c) Procedures for functional equipment level fault isolation; based on fault sensing indicators supplemented, as required, by test point measurements using built-in test equipment

(d) Equipment-level isolation techniques the use of which will permit identification of the problem area to a single module or replaceable part

(e) Routine tests, adjustments, alignment, and other "preventive" procedures which are performed at periodic intervals

(4) Ready Reference Documentation - This documentation is limited to that information routinely required by the technician in a given equipment repair area. The documentation should be easily usable in the work area - i.e., capable of being held with one hand, remaining open to a given section, permitting easy replacement or additions, and suitable for storage in the work area. It should contain only routine checkout, alignment, and preventive maintenance procedures; fault monitoring interpretation and replacement data; supplemental troubleshooting techniques required to complement the automatic fault detection and isolation system; and item and unit spare parts ordering data keyed to system identity codes.

11.5.4 Data Collection and Analysis (During Field Deployment)

A new system or equipment begins to accrue valuable experience data with its initial introduction into the field. These data, accurately recorded and consistently reported, provide the final basis for judging suitability of the system for continuing deployment. Thereafter, the reporting system can become the essential basis for an effective R&M feedback loop if provisions are made for continuous reporting and periodic analysis of maintenance experience data throughout the deployment phase and if formal procedures are established for progressive correction of discrepancies revealed by the analysis. On the other hand, if the reporting system is not fully exploited analytically and applied dynamically in a formal corrective action program, the R&M feedback loop is short circuited and serves no purpose other than logistic surveillance.

Data required to effectively assess, monitor, control and improve the R&M of fielded systems and equipment items include hours of operation (and appropriate data on operating characteristics), performance measures and assessments, application environmental factors, and, most important, failure and maintenance data. The feedback of information obtained from the analysis of failure during actual use is essential to reliability growth. The focus of the data collection should be on tracking failure modes, not symptoms.

Development of a formal and well-documented field data recovery, analysis and feedback system

is a key element in an effective R&M program. The data recovery and feedback program is designed to be compatible with and incorporate data from other data collection efforts during acquisition and storage. An effective data system provides output information that can be used for:

(1) R&M assessments

(2) R&M tracking

(3) Comparative analysis and assessments

(4) Determination of the effectiveness of R&M tasks and management concepts

(5) Identification of critical equipment, components and problem areas

(6) Compilation of historical component failure rates for design predictions

Plans are prepared that describe the specific mechanisms for collecting operation, maintenance and installation data at field sites, depots, and disposal areas as well as during factory test for feedback. Included are detailed instructions, document forms, and the delineation of responsibilities for implementation. Furthermore, the system must be planned such that it is compatible with standard military data systems. It should be noted that during acquisition the data system is primarily the responsibility of the system equipment developer where control by the military is established through reporting of summary data and deliverable data items.

During operation, military maintenance data collection systems are used to record and accumulate ongoing data. These programs, including the Army's TAMMS (The Army Maintenance Management System), the Navy's 3M and the Air Force's REMIS and other maintenance data collection systems, are primarily maintenance oriented. Maintenance actions are reported and processed in a computer data bank at three levels: equipment, assembly board, and piece-part. For each entry, the failure indicator is reported along with codes identifying such things as the base command and the equipment nomenclature. They do not, however, report operating time. Moreover, the field use environment and the field maintenance environment are not adequately quantified to ensure consistent interpretation of field data. Thus, field reliability cannot be assessed using data from only the military systems. In order to assess reliability and to compare the attained field reliability with that specified and estimated during acquisition, both equipment/system failure (or maintenance) data and their associated operating time(s) are required. The associated equipment/system operating time must be estimated or obtained directly from the operational units themselves. Operating times are recorded in station logs and the equipment inventory, with associated records of uptime, storage time and maintenance times, by month.

In addition to the previously mentioned maintenance data collection systems, the Department of Defense instituted the Reliability Analysis Center (RAC), a DoD Information Analysis Center, which functions as a focal point for the recovery of reliability test data and experience information on electronic, electrical, and electromechanical components, and R&M data on the equipments/systems in which these components are used. Reliability experience information is disseminated by the RAC through reliability data compilations, handbooks and appropriate special publications to upgrade and support system reliability and maintainability.

These publications cover the following:

(1) Nonelectronic Parts Reliability Data (NPRD)

(2) Nonoperating Reliability Databook (NONOP-1)

(3) Failure Mode/Mechanism Distributions (FMD)

The publications are updated and reissued periodically, deleting outdated data entries and incorporating new acquisitions from the latest technologies and applications. For additional information on the RAC, as well as other specialized DoD Information Analysis Centers, see Reference 9.

11.5.5 System R&M Assessment

Once an equipment/system is deployed, its R&M performance is periodically assessed based on the analysis of collected field operational/failure data as described in the previous section, as well as information derived from other sources. Programs have been established to assess R&M in a manner so as to yield consistent and accurate data and information that can be fed back into the product improvement process as well as to provide a “lessons learned” information base for subsequent acquisitions. The programs are designed to provide data and information that can be used to:

(1) Uncover problem areas, effect timely corrective action, and provide a solid basis for system R&M improvement programs.

(2) Determine the effectiveness of design, test and program concepts applied during system acquisition.

(3) Track the performance and, in particular, the R&M of the fielded system.

Application of the feedback loop to service evaluation of R&M and correction of R&M problems is accomplished in five major steps, the last of which becomes the first step in a repetition of the cycle:

(1) Acquisition of Required Data - Use the data collection and reporting system to acquire the basic service use experience data, supplemented as necessary by system configuration and engineering data, and operational information to ensure correlation between reported maintainability experience and the conditions under which the experience data was accrued.

(2) R&M Assessment - Analyze the reported experience data to derive a measure of the R&M parameters (e.g., failure rate, MTBF, mean corrective maintenance time ($\overline{M}_{ct}$), maximum corrective maintenance time ($M_{max_{ct}}$), maintenance manhours per operating hour, logistics delay time, etc.) at system, subsystem, equipment, major component, and lower levels, corresponding to the levels to which R&M was allocated, specified, and demonstrated during the development phase.

(3) Problem Definition - Identify, investigate, and describe the underlying problems which account for major discrepancies or deficiencies noted in the analysis of (2) above in terms amenable to translation into corrective action as design changes, documentation changes, maintenance or logistics procedural changes, etc., as appropriate. Introduce on a limited sampling basis such supplementary data recording forms, time clocks, instrumentation, and reporting instructions as required for the assessment of R&M where the field values greatly exceed predicted or demonstrated values.

(4) Corrective Action Assignment - Formally assign corrective action responsibility accompanied by problem descriptions developed under (3) above with specified criteria for verifying achievement of corrective action objectives.

(5) Follow-Through - Reassess R&M as in (2) above to evaluate effectiveness of corrective actions, to compare R&M trends relative to established improvement objectives, and to reevaluate problems identified in earlier assessments. This step begins the assessment cycle all over again.

Department of the Army, Readiness Command (DARCOM) Regulation 702-9 (Ref. [10]) defines the policies and procedures of a formal R&M System Assessment Program established by the Army. This regulation requires that assessments be performed in order to determine whether the fielded system has satisfied user needs for mission performance and logistic support. They are conducted in order to identify and take corrective action on problems which are degrading user satisfaction, operational readiness, and life cycle cost. Through the performance of such assessments the Army determines how a system is operating, uncovers and corrects problems in system operation and support, and thus helps achieve complete user satisfaction.

SECTION 11: PRODUCTION AND USE (DEPLOYMENT) R&M

As presently structured, the System Assessment Program includes the assessment of all aspects of fielded system operations including:

(1) Technical
- A narrative description of the system and its support equipment
- Original design objectives
- The results of development and operational tests
- Corrective action results

(2) Operational
- Initial field performance parameter values
- Changes incorporated into the fielded system (e.g., payload, accuracy, reliability, availability, and maintainability)
- Present field performance parameter values

(3) Environmental
- Individual component shelf-life values
- The reliability of components which require storage stockpile testing
- The effect stored components are having on overall system reliability

(4) Human Factors
- The user's opinion of the adequacy of the system
- The quantity of personnel, by military occupational specialty
- The quality of personnel, by military occupational specialty

(5) Support
- Current problems
- Development initiatives for replacement
- Effectiveness of the present logistic support system
- Improvement actions required
- System improvement plans

DARCOM Regulation 702-9 states that maximum use will be made of existing field data to assess these areas. Other data sources include

(1) Sample data collection programs

(2) Field visits and surveys

(3) User questionnaires

(4) User conferences

(5) Logistic personnel and field maintenance technicians

11.5.6 System R&M Improvement

In addition to optimizing R&M during acquisition through aggressive design, development, and production programs, substantial R&M growth potential exists during deployment. Some of this growth occurs naturally as the operations and maintenance personnel become more familiar with the equipment. However, to accelerate the growth rate and achieve significant increases in operational R&M requires the application of a closed-loop process of positive corrective action based on analysis and assessment of field R&M data. For newly deployed equipment, this closed-loop process can achieve significant reliability improvement, especially when used within the context of a total, disciplined system assessment program as discussed in the previous subsection. Reliability growth is based upon the iterative process of monitoring system operation to identify actual or potential sources of failures, to redesign out the failure source, and to fabricate and apply changes which improve system reliability. As such, reliability growth can be applied during development, production, or during operation. For fielded systems, the reliability growth process is a valuable tool to attain reliability improvements and achieve savings that could more than offset the cost of the reliability improvement program. The process is also performed during field deployment to eliminate problem areas not evident during the development phase.

The R&M improvement program must work in conjunction with the data collection and assessment programs (as discussed in the previous section) in a total integrated process consisting of data collection, system assessment and improvement selection, development, and implementation to achieve reliability growth in the field.

As described in more detail in the previous section, the program is an iterative feedback process consisting of the following steps:

(1) Acquisition of required data

(2) R&M assessment

(3) Problem definition

(4) Corrective action assignment

(5) Follow through to evaluate effectiveness of corrective action(s)

SECTION 11: PRODUCTION AND USE (DEPLOYMENT) R&M

The action of improving system reliability involves a systematic review of several concepts which appear from the backup data to be most useful for reliability cost tradeoff considerations, among which are:

(1) The reduction of failure rates by operating components at reduced (derated) stress levels, accomplished by selecting components which have ratings well in excess of those required for their system application.

(2) The use of improved components for which reliability has been significantly increased through special manufacturing techniques, quality control procedures, and testing methods.

(3) Design simplification to eliminate parts or components.

(4) The substitution of functionally equivalent items with higher reliability.

(5) The overall reduction of failure rate through increased control of the internal system environment, e.g., through reduction of ambient temperature, isolation from handling effects, and protection from dust.

(6) The provision of design features which enable prediction of incipient failures and permit remedial action to be taken before an operational failure occurs.

(7) The provision of design features which reduce the probability of human-initiated errors.

(8) The provision of multiple identical parts, paths or higher functional levels (redundancy) in order to prevent a system failure in the event that one element fails.

(9) The reduction of failure rate through increased control of the environment external to the equipment, as through reduction of ambient temperature, isolation from handling effects, isolation of operator from ambient noise, and protection of equipment from dust.

(10) The implementation of controlled screening and burn-in tests for the purpose of significantly reducing incipient failures due to undetected defects in workmanship or components.

Similarly, maintainability (MTTR) can be improved by incorporating improved use of maintenance practices, providing higher quality technical manuals and maintenance aids or possibly better training to improve the skill level of the technicians.

Computing the impact of the improvement recommendations which appear most useful for cost tradeoff consideration on MTBF, MTTR, overall downtime and system performance, using the

methods and techniques previously described, and determining the total cost for their implementation is an essential step in evaluating the effectiveness of the improvement.

Critical to the analysis process is the ability to assess quantitatively the cost effects of reliability and maintainability. The cost of each recommended change must take into account total cost throughout the life cycle of the system and accordingly must include cost elements associated with design, manufacture, procurement, installation, and field use (i.e., operation, maintenance, and logistics).

The final step is to compute cost/benefit factors, i.e., develop a numeric for each R&M recommendation which reflects the total cost of the change, its impact on system performance, and the cost avoidance to be realized over a given time period by their implementation. This will allow the determination of those change recommendations which have maximum cost effectiveness. (See Section 8.1 for a discussion on reliability data collection and analysis). The recommended changes can then be presented in an improvement plan in prioritized order of cost effectiveness, as defined by the computed cost/benefit factors.

11.6 References For Section 11

1. Schafer, R.E., A.E. Sari and S.J. Van DenBerg, Stress Screening of Electronic Hardware. RADC-TR-82-87, May 1982, (AD-A118261).

2. Environmental Stress Screening Guidelines. The Institute of Environmental Sciences, Library of Congress Catalog Card No. 62- 38584, 1981.

3. Navy Manufacturing Screening Program. NAVMAT P-9492, Naval Material Command, May 1979.

4. Care of Supplies in Storage (COSIS). Army Regulation AR 740-3, 1993.

5. Product Assurance - Storage Serviceability Standards (SSS). Army Regulation DARCOM-R 702-23.

6. Product Assurance Depot Quality Assurance System. Army Regulation AMC-R 702-7.

7. Army Supply Bulletin SB740-99-1, Storage Serviceability Standard for TSARCOM Material.

8. Maintenance of Supplies and Equipment, AMC Guide to Logistics Support Analysis. AMCP 750-16, Headquarters, Army Materiel Command, January 1978.

9. Directory of the Department of Defense Information Analysis Centers. Defense Technical Information Center, Defense Logistics Agency, Cameron Station, Alexandria, VA 22304-6145, April 1995.

10. DARCOM Regulation 702-9, Department of the Army, September 1977.

11. Environmental Stress Screening Guidelines. Tri-Service Technical Brief, 002-93-08.

12. Environmental Stress Screening Guidelines for Assemblies. 1984 & 1988, The Institute of Environmental Sciences.

13. Environmental Stress Screening Guidelines for Assemblies. 1990, The Institute of Environmental Sciences.

14. Environmental Stress Screening Guidelines for Parts. 1985, The Institute of Environmental Sciences.

15. Impact of Nonoperating Periods on Equipment Reliability, RADC-TR-85-91.

16. Kinlaw, Dennis C., Continuous Improvement and Measurement for Total Quality: A Team-Based Approach, Pfeiffer & Company/Business One Irwin, 1992.

17. Wilson, Lawrence, H., Eight-Step Process to Successful ISO 9000 Implementation: A Quality Management System Approach, ASQC Quality Press, Milwaukee, WI, 1996.

18. Prescott, Jon, “What the $75 Books Don’t Tell you about ISO 9000 Documentation,” Reliability Review, Volume 15, Number 2, June 1995.

19. Breitenberg, Maureen, “Questions and Answers on Quality, the ISO 9000 Standard Series, Quality System Registration and Related Issues,” NISTIR 4721, National Institute of Standards and Technology, April 1993.

12.0 RELIABILITY MANAGEMENT CONSIDERATIONS

12.1 Impacts of Acquisition Reform

As discussed in Section 4.0, recent Acquisition Reform policies have resulted in the elimination of many historical reliability standards from the DoD procurement process. Past versions of this handbook heavily relied on MIL-STD-785 (canceled 30 July 1998), *Reliability Program for Systems and Equipment Development and Production,* to explain and provide guidance on the makeup, planning and management of a reliability program. However, under new reforms in acquisition, such standards can no longer be levied as a requirement on the system development contractor. In the past, the procuring agency was able to develop a statement of work (SOW) that specifically stated the contractor was to develop a Reliability Program Plan, and further, which reliability tasks from MIL-STD-785 (canceled 30 July 1998) were targeted to be performed to meet stated quantitative reliability requirements for the system to be procured. Now, as part of the cited reform policies, MIL-STD-785 has been canceled as of 30 July 1998, and military standard documents, with some exceptions, may not be imposed without a waiver. On the other hand, there is nothing in the latest acquisition reform language that prevents the system developer from proposing to use any current or previously existing military standard or handbook as the basis for implementing a design approach or program as part of an overall development approach.

12.1.1 Acquisition Reform History

On June 29, 1994, Secretary of Defense William Perry issued a five-page memorandum, “Specifications & Standards - A New Way of Doing Business.” The intent of the memorandum can be summarized as three "overarching” objectives:

(1) Establish a performance-oriented solicitation process

(2) Implement a document improvement process

(3) Create irreversible cultural change in the way DoD does business

The DoD is working to streamline the way in which procurement is managed and to adopt commercial practices whenever possible. It is reassessing and trying to improve the way it does business to decrease costs and increase customer satisfaction.

As will be explained, military standards and specifications may be cited for guidance in a Department of Defense solicitation but **shall not** be cited as requirements unless a waiver is granted. Commercial standards may be cited for guidance. Although not specifically prohibited by policy at the time this handbook was written, commercial standards should not be mandated as requirements. Given the spirit of the new acquisition policy, mandating a commercial standard is no different than mandating a military standard. In either case, the procuring agency would be telling the bidding contractor what to do and how to do it, at least to the extent that the

cited standard provides suggestions on the tasks and activities needed for reliability. **The main objective of the new policy is to use performance specifications.** Only when performance specifications are inadequate for fully describing what the government wants should commercial specifications and standards be considered. And only when commercial specifications and standards are inadequate should a waiver to use a military specification or standard be considered.

12.1.1.1 Performance-based Specifications

(1) A performance specification states requirements in terms of the required results and provides criteria for verifying whether or not the requirements have been met. Performance specifications do not state the methods for achieving the required results. They have the following characteristics:

(a) Requirements should be stated quantitatively

(b) Requirements should be verifiable

(c) Interfaces should be stated in sufficient detail to allow interchangeability with parts of a different design

(d) Requirements should be material and process independent

(2) There are four types of performance specifications: commercial item descriptions (CIDs), guide specifications (GSs), standard performance specifications (SPSs), and program-unique specifications.

(a) Commercial Item Descriptions. An indexed, simplified product description prepared by the government that describes, by performance characteristics, an available, acceptable commercial product that will satisfy the Government's needs. Guidance for CIDs is given in the General Services Administration Federal Standardization Manual (Chapter 6), in the Defense Standardization Manual, DoD 4120.3-M, and in DoD 5000.37-H. By definition, CIDs are only to describe requirements in terms of function, performance, and essential form and fit requirements. CIDs are listed in the DoD Index of Specifications and Standards (DoDISS).

(b) Guide Specifications. Guide specifications identify standard, recurring requirements that are common for like systems, subsystems, equipments, and assemblies. The format of a GS forces the user to tailor the document to the specific application. Guidance for GSs is in DoD 4120.3-M. GSs are listed in the DoD Index of Specifications and Standards (DoDISS).

(c) Standard Performance Specifications. A specification that establishes

requirements for military-unique items used in multiple programs or applications. MIL-STD-961 includes guidance on the format and content of SPSs.

(d) Program-Unique Specifications. This type of specification, also called a system specification, establishes requirements for items used for a particular program or weapon system. Little potential exists for using these specifications in other programs or applications. They should be performance-based but may include a blend of performance and detail design requirements. They are restricted to items for which the preceding categories of performance specifications are not applicable.

(3) Performance specifications are also categorized by the type of item being acquired. Those used to acquire materials are called material specifications, to acquire components are called component specifications, and to acquire systems are called system specifications. The Department of Defense has issued a guide to performance specifications, SD-15 (Ref. [1]). Issued under the Defense Standardization Program, the guide covers the writing of performance requirements, standard performance specifications, guide specifications, and program-unique specifications. The discussions under a and b above are based on SD-15.

12.1.1.2 Other Standardization Documents

(1) Standards. There are four types of standards: interface, test method, manufacturing process, and practices.

(a) Interface Standards. An interface standard is one that specifies the physical or functional interface characteristics of systems, subsystems, equipments, assemblies, components, items, or parts to permit interchangeability, compatibility, or communications. Waivers **are not required** to use military interface standards as requirements in Department of Defense solicitations.

(b) Test Method Standard. A test method standard is one that specifies procedures or criteria for measuring, identifying, or evaluating qualities, characteristics, and properties of a product or process. Military test method standards **shall not** be cited as requirements in a Department of Defense solicitation unless a waiver is granted.

(c) Manufacturing Process Standard. This type of standard states the desired outcome of manufacturing processes or specifies procedures or criteria on how to perform manufacturing processes. Military manufacturing process standards **may not** be cited as requirements in a Department of Defense

solicitation unless a waiver is granted.

(d) Standard Practice Standard. A standard practice standard is one that specifies procedures on how to conduct certain functions or operations. These procedures are not related to manufacturing processes. It has not yet been decided if standard practice standards may be cited as requirements in a Department of Defense solicitation without a waiver.

(2) Handbooks. A handbook is a guidance document that provides engineering or technical information, lessons learned, possible options to resolve technical issues, classification of similar items, interpretive direction and techniques, and other types of guidance or information. The purpose is to help the customer or the seller to design, construct, select, manage, support, or operate systems, products, processes, or services. Military handbooks **shall not** be cited as a requirement in a Department of Defense solicitation, contract, specification, standard, drawing, or any other document.

12.1.1.3 Overall Acquisition Policy and Procedures

The primary documents governing defense acquisition are DoD Directive 5000.1 and DoD Regulation 5000.2-R. Both documents were revised as a result of Defense Acquisition Reform. A third document, DoD 5000.2-M has been canceled. The revisions of 5000.1 and 5000.2-R incorporate new laws and policies, separate mandatory policies and procedures from discretionary practices, and integrate acquisition policies and procedures for weapon systems and automated information systems. In addition to the two documents, an Acquisition Deskbook is available to DoD procuring activities. The Deskbook is an automated repository of information consisting of a Desk Reference Set, a Tool Catalog, and a forum for information exchange. The Reference Set consists of mandatory Guiding Principles, discretionary Institutionalized Knowledge, and Sage Information (expert wisdom and lessons learned). Information about the Acquisition Deskbook can be obtained using the Internet:

<http://deskbook.osd.mil/deskbook.html>.

The major themes of the new acquisition documents are teamwork, tailoring, empowerment, cost, commercial products, and best practices. These themes can be summarized as follows: acquisition should be a team effort among all concerned in the process, the acquisition approach for a specific system should be tailored based on risk and complexity, acquisition will be conducted with a customer focus, cost will be an independent variable in programmatic decisions, commercial products should be used when practical, and acquisition is now more closely modeled on best commercial business practices.

12.1.1.4 Impacts on Reliability Management

Despite the recent changes in Acquisition Reform policy, reliability management methods and concerns have not changed dramatically. The major change is in how the reliability program tasking is defined, and the greater emphasis on the use of Commercial-Off-The-Shelf (COTS)

and Nondevelopmental Item (NDI) equipment. It is now the contractor or supplier who has to decide what needs to be done to cost effectively achieve a stated reliability capability. Further, the government or other customer must evaluate which, of potentially many different approaches provides the best value. As will be discussed in this section, it is still important to the contractor to develop a reliability program plan and manage reliability as an integral part of the product design and development effort. For the customer, greater emphasis must be put on defining the required levels of reliability, availability and maintainability that are needed to meet performance expectations. This will include defining product usage profiles, the maintenance concept, operating and maintenance environments, and other life cycle factors such as storage and handling conditions. This information is essential if the contractor is to define a reliability program that meets stated requirements within cost and schedule constraints.

12.2 Reliability Program Management Issues

In managing a reliability effort, whether as a customer or as a supplier, there are several key issues that must be addressed. For any product or system, the key issues from any customer's perspective are:

(1) What measures of reliability are important?
(2) What levels of reliability are necessary to meet my needs?
(3) How will it be ensured that the required levels of reliability have been achieved?
(4) What reliability activities are the most effective for the product or system, such that the reliability program objective is achieved? Note: Even when the contractor selects the reliability activities, program offices must be able to judge which activities are applicable to their particular acquisition. Such judgement allows the acquisition staff to determine the risks associated with a contractor's proposed effort and, if necessary, negotiate changes.

From a supplier's perspective, the key issues are:

(5) What reliability activities are the most effective for the product or system, such that the reliability program objective is achieved?
(6) What reliability design goals are appropriate to ensure that customer's needs are met?
(7) What design approaches will be most effective in achieving the required reliability in the expected environmental and usage profiles?
(8) What tasks can be effectively used to assess progress towards reliability goals and requirements?
(9) What are the most appropriate means to determine if the reliability objective has been achieved?
(10) How can the designed-in reliability be retained during manufacturing and operational use, thereby ensuring reliable performance?

Each of the above issues must be addressed as part of meeting the basic objectives of product reliability which are: understanding the customer's requirements, meeting the requirements and demonstrating the requirements have been met.

In a commercial world, the customer is not usually concerned with the second set of issues - they are left to the seller to confront. If the seller does a poor job, the customer will go elsewhere for the product. Thus, competition in the marketplace provides a strong incentive to "do it right." In the military world, the level of competition is often much lower than in the commercial world. If dictated by the nature of the product (e.g., used only by the military), the risks (e.g., very high with unproved technologies being used), and the type of acquisition (e.g., totally new development), it will be necessary for the government customer to take more of an active role in addressing the second set of issues. (Some industrial customers also may be involved with the second set of issues, especially those dealing with measuring progress and determining the achieved level of design reliability). The form that this role takes, however, has changed.

Previously, by imposing standards and specifications, the military customer could force contractors to use certain analytical tools and methods, perform certain tests in a prescribed manner, use parts from an approved list, and so forth. The objective under Defense Acquisition Reform is not to tell contractors how best to design and manufacture a product. The responsibility for making such decisions has shifted from the government to the contractor. None-the-less, military customers are still more likely to be aware of the second set of issues than are commercial customers. Consequently, specifications issued by the government will probably continue to be more detailed than those issued by commercial organizations. Of course, when COTS products or non-developmental items (NDI) (Ref. [2]) are being procured, a more commercial approach to acquisition by the government is appropriate.

12.3 Reliability Specification Requirements

It is through the solicitation that a customer describes a needed product and solicits bids from competing sources to develop the product. Typically, a solicitation consists of a specification and a statement of objectives (SOO) or statement of work (SOW). (Note: Military solicitations must be issued in accordance with the Federal Acquisition Regulations).

(1) The specification should be a performance specification, one that states requirements in terms of the required results with criteria for verifying compliance but does not state the methods for achieving the required results.

Traditionally, a military or commercial acquisition has only one specification. Some customers, however, have adopted a new approach to specifications. They issue an initial specification and then work with each prospective bidder to develop a specification unique to that bidder. In that way, multiple specifications are developed. The specifications reflect the technical capability of each bidder, and one bidder's specification may be more demanding than others, although all must meet the customer's needs. The bidder whose specification and price represents a best-value is awarded the contract.

In some cases, the customer does not provide a specification. For example, the general public does not provide automobile manufacturers with specifications for a vehicle. Instead, the automobile manufacturers must develop their own specifications based on such considerations as: federal, state, and other government laws and regulations, benchmarking of competitors' products or market surveys and opinion polls.

(2) The SOW normally includes constraints, assumptions, and other criteria that the bidders must consider in developing and manufacturing the product. For example, the customer should identify how the product will be used (operating concept) and supported (support concept).

The SOW may also include specific activities or tasks required by the customer. In the past, the SOW included with a military solicitation almost always identified specific tasks, such as "perform a Failure Modes and Effects Analysis." As stated earlier, the approach under Defense Acquisition Reform is to allow the bidders to identify planned activities and to explain why, how, and when these activities will be performed. Commercial customers seldom specify specific tasks but are, of course, free to do so.

Instead of the traditional SOW, some procuring agencies use a statement of objective (SOO). Considered more in keeping with the spirit of acquisition reform, the SOO is concise and written to allow the contractor as much flexibility as possible in responding to the solicitation. A typical SOO has five sections: Objective of the Program (Solicitation), Objective (Purpose) of the Contract, Scope of the Contract, Work to be Accomplished under the Contract, and Program Control. The SOO is included as an attachment to an RFP, typically appended to Section L. Normally, the government will ask offerors in response to the SOO to prepare and provide a SOW in their proposals. Specific efforts defined in an offerors SOW shall be traceable to the SOO.

12.3.1 Template for Preparing Reliability Section of Solicitation

In developing the reliability portion of a procurement package, two distinct areas must be covered. These areas are performance-based requirements and programmatic and reporting requirements.

Performance-based requirements for reliability that may be placed in a specification include but are not limited to: Probability of Success P(S), Mission Reliability $R(t_m)$ or MTBF. In the case of programmatic and reporting requirements, the customer may require the seller to prepare and submit reports describing the results of analyses, tests, and other activities conducted by the contractor and described in the reliability program plan to design and manufacture a reliable product.

For NDI and COTS items the customer may require the seller to furnish operational data and the results of testing to substantiate reliability claims. In addition, the customer may require the seller to propose a method for verifying that reliability requirements have been met.

It should be the supplier's responsibility to select the tasks and other activities that will achieve these objectives and to describe the tasks and activities in the reliability program plan. When the customer mandates specific activities, previously referred to as tasks, the contractor is, to some extent, relieved of the responsibility to ensure each activity is value-added and preferable to other activities.

The following Template provides an outline for developing the reliability portion of a procurement package. The following conventions are used.

> Words within { } pertain only to new development efforts; words within [ ] pertain only to procurement of NDI or COTS. Procurement packages for programs involving both NDI/COTS and new development items should address each type of item separately but require that the reliability efforts be integrated.
>
> Blanks __ indicate where the user of the template must provide a value or other information.
>
> *Italicized words* are optional instructions that may or may not be used depending on the desires of the user and the needs of the procurement.
>
> Notes to the reader are in parentheses with NOTE printed all in caps.

The reader is reminded that when purchasing NDI or COTS, the best course of action may be to require only data that substantiates any claims for performance and to emphasize the role of manufacturing processes (for NDI not yet in production) in determining the reliability of the product. In some cases, even that data may not be needed if either the customer has already determined (through its own testing of samples, for example) that the product has the requisite performance or if use or independent testing of the product in actual applications has shown the product's performance to be satisfactory (for example, a personal computer in an office environment).

As previously discussed, in lieu of issuing a SOW with a specification, some customers now issue a SOO and require the offerors to include a SOW as part of their proposals. The SOO could include reliability objectives for the acquisition program, such as those listed in Section 3 of this Handbook. The best manner to respond to the solicitation would be left entirely to the bidders (for example, whether or not to have a reliability program plan).

A draft solicitation can be released by a customer for comment and suggestions for a statement of work by potential bidders. Based on the comments and suggestions received, a "negotiated" statement of work reflecting the bidders' best ideas on achieving the required level of reliability would be included in the formal solicitation (assuming a SOO is not being used instead).

TEMPLATE FOR DEVELOPING RELIABILITY PORTION OF A PROCUREMENT PACKAGE

SECTION L

(NOTE: Not all possible requirements are listed, and not all listed requirements are necessarily applicable to all procurements).

1. The bidder shall describe how the reliability requirements of the solicitation will be met. If a bidder elects to submit a reliability program plan, the plan may become part of the contract upon contract award. In any event, the bidders' responses will be evaluated using the following criteria.

1.1 The bidder shall describe all activities considered to {be necessary for ensuring the development of a} [have contributed to designing and manufacturing a] reliable product. For each activity, the bidder shall describe the objective, rationale for selection, method of implementation, methods of assessing results, and any associated documentation.

1.2 The bidder shall explicitly address how the included activities {will be} [were] integrated into the product and manufacturing design processes.

1.3 The bidder shall show how the results of the included activities {will be} [were] used to support other activities, such as logistics planning, safety analyses, etc.

1.4 The bidder shall explicitly show a clear understanding of:

a. the importance of designing in reliability and the relationship of reliability to other system performance characteristics.

b. reliability design techniques, methodologies, and concepts

c. the importance of integrating reliability activities into the overall systems engineering process

1.5 The bidder shall show how the following objectives {will be} [were] met:

a. thoroughly understand the design

b. validate the design and manufacturing processes

c. ensure proper parts application

d. address all portions of the product including those provided by suppliers and vendors

e. evaluate the achieved reliability

{f. determine feasibility}

TEMPLATE FOR DEVELOPING RELIABILITY PORTION OF A PROCUREMENT PACKAGE

STATEMENT OF WORK

1. The bidder shall identify all work activities {to be} conducted to meet the reliability performance requirements cited in _____ . In so doing, the bidder shall:

- *identify the specific objective of each work activity*
- *identify each work activity {is to be} [was] conducted*
- *identify specific product or outcome {expected} [achieved]*
- *explain how these work activities fit into the overall design effort*
- *identify any standards (commercial, military or company) that {will be} [were] used in performing the work activities*

1.1 The bidder will identify special reliability risks or issues associated with the chosen design approach and describe which work activities[ed] address these risks or issues and how.

1.2 {The bidder will identify work activities that are new to the company or are being used for the first time and explain what steps will be taken to minimize any risk associated with first use).

NOTE: Regarding the next section, the reader is reminded that mandating tasks, even for new development, is somewhat risky because it relieves the bidders of the responsibility for selecting the best means to accomplish the desired ends (in this case, meet the reliability performance requirements). Mandating tasks should be done only after careful consideration of the advantages and disadvantages of doing so. **Even then, bidders should not be told how to accomplish the required task. And, unless a waiver is obtained, processes may not be contractually mandated (reference OUSD (A&T) memorandum dated 18 September 1997, "Requiring Processes on Contract.")**

2. The following activities will be conducted by the bidder and reflected in the technical approach.

2.1 Implement a Failure Reporting and Corrective Action System (FRACAS).

2.2 Conduct either a Fault Tree Analysis (FTA) or Failure Modes and Effects Analysis (FMEA). Rationale for selecting one over the other will be given.

2.3 Institute an Environmental Stress Screening program. The bidder should indicate how the stresses and stress levels will be determined.

2.4 Develop a reliability model and make initial reliability predictions using that model. All predictions should be made at a stated level of confidence.

2.5 Implement a parts control program. Parts will be derated; the bidder will indicate how derating criteria will be developed.

TEMPLATE FOR DEVELOPING RELIABILITY PORTION OF A PROCUREMENT PACKAGE

STATEMENT OF WORK

2.6 Conduct thermal analyses to ensure parts and components are not subjected to thermal stresses that exceed design tolerances.

2.7 Conduct formal reliability growth testing for the purpose of uncovering design deficiencies and other failure mechanisms.

2.8 Conduct a reliability demonstration. The bidder shall explain how the demonstration will be implemented and the underlying statistical basis of the demonstration.

2.9 Conduct a _____ (NOTE: Others as determined by buyer).}

(NOTE: All reports, data requirements, and deliverable documents should be identified in the Contract Deliverables Requirements List (CDRL). Data items can include FMEA results, results of trade studies, thermal analyses results, and so forth. Data items should be selected based on the nature of the development, the level of risk, intended use of the item [benefit], and cost. The CDRL should provide data format and content preparation instructions and data delivery requirements. Although the text of the SOW should not include these items, a data item description number listed in the CDRL may be cross-referenced in the SOW. This cross reference should usually be made in the paragraph describing the task that will lead to the development of the data or document).

TEMPLATE FOR DEVELOPING RELIABILITY PORTION OF A PROCUREMENT PACKAGE

THE SPECIFICATION

(NOTE: User should select the life units most appropriate for each product. For example, operating hours might be the best measure for an engine, miles traveled for a truck, cycles for a starter, and so forth).

1. The bidder shall carry out the activities described in the Statement of Work to achieve the following levels of reliability. Note: All values are the minimum acceptable values at a _____ ___ confidence level.

1.1 The product shall exceed _____ life units between any failure that requires a maintenance action

1.2 The product shall exceed _____ life units between any failure that prevents the product from performing its mission

2. The service life of the product will be _____ life units. Service life is defined as the period over which the product can be operated and maintained in accordance with the contractor's prescribed maintenance and operating procedures before continued use becomes prohibitively expensive or risky without major structural repair or replacement, system modifications or replacement, or other actions not considered normal day-to-day maintenance and upkeep.

3. *The product will be designed so that its reliability and service life will not be reduced due to the effects of being shipped by land, sea, or air or by periods of storage up to _____ life units.*

4. All reliability requirements apply to the product as it will be used in the environment defined in Section _____ of the Specification and in accordance with the operating and support concepts defined in Section ____ of the _____. (Customer must indicate where this information is provided in the solicitation).

5. Other. (Customer should indicate other requirements or information pertinent to the required level of reliability).

TEMPLATE FOR DEVELOPING RELIABILITY PORTION OF A PROCUREMENT PACKAGE

STATEMENT OF OBJECTIVES

1.0 Program Objectives

a. The program is: (here the customer defines the program as: (1) multi-phased, (2) single-phase, or (c) one program with multiple contractors).

b. The objective of the program is to design, test, and manufacture (*) to satisfy the performance requirements of the specification to meet a need date of [date].

2.0 Contract Objectives. The contractor shall meet the following objectives.

2.1 Design, Analysis, and Test

Design the [*] to satisfy performance requirements as defined in [cite applicable section of RFP]. Perform such analysis and tests necessary to design the [*], to reduce risk, and to verify that the product meets all performance requirements.

2.2 Configuration Management

Establish a product baseline to define the configuration of the [*] with a verified capability to satisfy all performance requirements. Establish and maintain a management process to thereafter control the product's configuration for the life of the contract. Document the design of the product baseline through the use of engineering data.

2.3 Quality Control

Institute a quality program to ensure the [*] is produced in accordance with engineering data, measuring and test equipment are properly maintained, and that appropriate actions are taken for nonconforming materials.

2.4 Logistics

Develop and deliver all data necessary to support the [*] (including provisioning, installation, and reprocurement data and operating and repair manuals) consistent with the maintenance concept as stated in [cite applicable section of RFP]. All data shall be in a form and format compatible with existing government data systems.

*Name of the product