Enterprise Reliability, Availability, Maintainability and Testability Plan

Document Change Record

Version Number	Change	Date
1.0	Initial RAMT Plan	

Contents

Reliability, Availability, Maintainability and Testability Program Plan

1. Introduction

A formal Reliability, Availability, Maintainability and Testability (RAMT) Program Plan is essential for achieving high levels of reliability, testability, maintainability and the resulting system availability and is required to be developed during any Project system development phase (if the Project goes through that phase) and refined over all additional life-cycle phases the Project is contracted to accomplish. However, a RAMT Program Plan must be developed at any Project life cycle phase if required by contract or is necessary to successfully accomplish the Project work efforts. It specifies not only what the reliability systems engineer does, but also the tasks performed by other participants (engineers, reliability analyses, etc. Each Project Reliability Program process developed according to the Enterprise Plan is to be approved by Project Management and the Project Systems Engineer responsible for allocation of resources since resource determination for manpower and budgets for testing and other tasks is critical for a successful program and in general, the amount of work required for implementation of an effective reliability program for complex systems is large.

The Enterprise systems engineering is a logically sequenced, consistent set of technical activities that translate a customer's needs and requirements into a balanced solution. Unfortunately RAMT predictions often do not provide the balanced solution systems engineering design analysis strives to obtain. RAMT prediction is any method used to assess the level of RAMT that is potentially achievable, or being achieved, at any point. Achieving metrics via a RAMT prediction will not ensure that the best system design is developed. Too often the following is forgotten about RAMT predictions:

1. RAMT predictions are a process, not a one-time activity, which should begin in early development and continue throughout the life of the system, with different methods used.
2. No one method of RAMT prediction is right for every item at all times. The "right" method is the one for which the requisite data are available and that is appropriate for the intended use of the RAMT prediction (i.e., comparison, spares computations, contractual verification, part characterization, system field performance, etc.).
3. RAMT predictions stated at a confidence level are more meaningful than a point estimate.
4. An understanding of the method itself, the maturity of the design and the fidelity of the data must temper the results of any method used to perform RAMT predictions

The Enterprise Systems Engineering attempts to ensure that the solution that satisfies the requirements of a RAMT prediction will also be the best overall solution with regards to multiple programmatic and technical considerations. Systems engineering expands the evaluation criteria to select criteria that best balance program requirements, such as system performance, total ownership cost, schedule, supportability, risk, etc. The criteria are selected based on the stated problem as well as the level and complexity of the analysis required.

1.1 Scope

This Enterprise RAMT Program Plan documents exactly what "best practices" (tasks, methods, tools, analyses and tests) are required for a generic system. Individual Project Reliability Processes define provided customer requirements for reliability assessment and how these methods, tools, analyses and test are to be accomplished. The overall scope addresses reliability, availability, maintainability and testability of the Enterprise developed/produced systems in a total context. This RAMT Program Plan (and any associated Project RAMT or specific Reliability Processes/Procedures) is also to be used to evaluate and improve the availability of a system by focusing on increasing testability and maintainability and provides a strategy for availability control. Note that each Project RAMT Process/Procedure must address overall system RAMT in context of the customer needs and contractual requirements.

This document covers the following major elements:

1. Interrelationships between reliability, availability, maintainability, and testability
2. Reliability design, analysis and prediction
3. Maintainability design, analysis and prediction
4. Availability design, analysis and prediction
5. Testability design, analysis and predictions
6. Failure Modes, Effects and Criticality Analysis (FMECA)
7. Failure Reporting and Corrective Action System (FRACAS)
8. Fault Tree Analysis (FTA)

1.2 Objectives

The objectives of the Enterprise RAMT Engineering, in the order of priority, are:

1. To apply engineering knowledge and specialist techniques to prevent or to reduce the likelihood or frequency of failures in our systems.
2. To identify and correct the causes of failures that do occur, despite the efforts to prevent them.
3. To determine ways of coping with failures that do occur, if their causes have not been corrected.
4. To apply methods for estimating the likely reliability of new designs, and for analyzing reliability data.

The reason for the priority emphasis is that it is by far the most effective way of working, in terms of minimizing costs and generating reliable, maintainable and available products. Note also that RAMT analyses and tasks are instrumental in supporting Project Safety Engineering functions. These, like Logistics Engineering, include reliability analysis such as the Failure Modes and Effect Criticality Analysis (FMECA) and RAMT Predictions. The output of these analyses serves as inputs to the Project safety analyses.

2. Interrelationships Between Reliability, Availability, Maintainability, and Testability

This Reliability Program Plan (and all associated Project Reliability Program Processes) can be used to evaluate and improve the availability of an Enterprise-developed/produced system by focusing on increasing testability and maintainability as well as reliability. Note that improving maintainability is generally easier than improving reliability. Maintainability estimates (repair rates) are also generally more accurate. However, because the uncertainties in the reliability estimates are in most cases very large, it is likely to dominate the availability (prediction uncertainty) problem; even in the case where maintainability levels are very high. When reliability is not under control more complicated issues may arise, like manpower (maintainers / customer service capability) shortage, spare part availability, logistic delays, lack of repair facilities, extensive retro-fit and complex configuration management costs and others. The problem of unreliability may be increased also due to the "domino effect" of maintenance-induced failures after repairs. But also note that only focusing on maintainability is not enough. If failures are prevented, none of the others are of any importance and therefore reliability is generally regarded as the most important part of availability. Reliability needs to be evaluated and improved related to both availability **and** the cost of ownership (due to cost of spare parts, maintenance man-hours, transport costs, storage cost, part obsolete risks, etc.). But, as GM and Toyota have discovered, Total Cost of Ownership (TCO) also includes the down-stream liability costs when reliability calculations do not sufficiently or accurately address customers' personal and equipment risks. Often a trade-off is needed between the two. There might be a maximum ratio between availability and cost of ownership. Note that whether availability or TCO is more important depends on the use of the system. For example, a system that is a critical link in a production system is normally allowed to have a very high TCO if this translates to even a minor increase in availability, as the unavailability of the system would result in a massive loss of revenue which can easily exceed the high cost of ownership. Testability of a system is also being addressed in the plan as this is the link between reliability and maintainability. The maintenance strategy can influence the reliability of a system (e.g. by preventive and/or predictive maintenance), although it can never bring it above the inherent reliability.

Note that it is possible (likely) that too much emphasis can be given to the prediction of reliability parameters that actually the effort devoted to the prevention of failure (reliability improvement). Failures can and should be prevented in the first place for most cases. The emphasis on quantification and target setting in terms of (e.g.) MTBF might provide the idea that there is a limit to the amount of reliability that can be achieved. In theory there is no inherent limit and higher reliability does not need to be more costly in development. Also note that prediction of reliability based on historic data can be very misleading, as a comparison is only valid for exactly the same designs, products, manufacturing processes and maintenance under exactly the same loads and environmental context. Even a minor change in detail in any of these could have major effects on reliability. Furthermore, normally the most unreliable and important items (most interesting candidates for a reliability investigation) are most often subjected to many modifications and changes. Engineering designs are in most industries updated frequently. This is the reason why the standard (reactive or proactive) statistical methods and processes are not as effective for engineering. Another note is that to be able to accurately predict reliability by testing, the exact mechanisms of failure must have been known in most cases and in most cases –

can be prevented. Following the incorrect route by trying to quantify and solving a complex reliability engineering problem in terms of MTBF or Probability and using the reactive approach is bad practice.

For existing systems, the responsible Project may directly analyze and try to correct the root cause of discovered failures and thereby render the initial MTBF estimate fully invalid as new assumptions (subject to high error levels) of the effect of the patch/redesign must be made. Another practical issue that needs consideration concerns a general lack of availability of detailed failure data and not consistent filtering of failure (feedback) data or ignoring statistical errors, which are very high for rare events (like reliability related failures). Very clear guidelines must be present in each Project Reliability Process to be able to count and compare failures, related to different type of root causes (e.g. manufacturing, maintenance, transport or system induced or inherent design failures). Comparing different type of causes may lead to incorrect estimations and incorrect business decisions about the focus of improvement.

To perform a proper quantitative reliability prediction for systems may be difficult and may be very expensive if done by testing. On part level, results can be obtained often with higher confidence as many samples might be used for the available testing financial budget, however unfortunately these tests might lack validity on system level due to the assumptions that had to be made for part level testing. The general conclusion that has been drawn from DOD and industry best practices is that an accurate and an absolute prediction - by field data comparison or testing - of reliability is in most cases not possible. **The introduction of MIL-STD-785 notes that reliability predictions should be used with great caution if used for anything other than comparison in trade-off studies.**

2.1 Factors Affecting RAMT

Many factors are important to RAMT: system design; manufacturing quality; the environment in which the system is transported, handled, stored, and operated; the design and development of the support system; the level of training and skills of the people operating and maintaining the system; the availability of materiel required to repair the system; and the diagnostic aids and tools (instrumentation) available to them. All these factors must be understood to achieve a system with a desired level of RAMT. During pre-systems acquisition, the most important activity is to understand the customer’s needs and constraints. During system development, the most important RAM activity is to identify potential failure mechanisms and to make design changes to remove them. During production, the most important RAMT activity is to ensure quality in manufacturing so that the inherent RAMT qualities of the design are not degraded. Finally, in operations and support, the most important RAMT activity is to monitor performance in order to facilitate retention of RAMT capability, to enable improvements in design (if there is to be a new design increment), or of the support system (including the support concept, spare parts storage, etc.).

2.2 Importance of RAMT

Achieving specified levels of RAMT for a system is important for many reasons; specifically the affect RAM has on readiness, system safety, mission success, total ownership cost, and logistics footprint. Corporate experience as well as DoD experience over the past decade suggests the following reasons why systems fail to achieve RAMT requirements:

1. Poorly defined or unrealistically high RAMT requirements.
2. Lack of priority on achieving Reliability and Maintainability
3. Too little engineering for RAMT. Among engineering process failures, these stand out:
    a. Failure to design-in reliability early in the development process.
    b. Inadequate lower level testing at component or subcomponent level.
    c. Reliance on predictions instead of conducting engineering design analysis.
    d. Failure to perform engineering analyses of Commercial-Off-The-Shelf (COTS) equipment.
    e. Lack of reliability improvement incentives.
    f. Inadequate planning for reliability.
    g. Ineffective implementation of Reliability Tasks in improving reliability.
    h. Failure to give adequate priority to the importance of Integrated Diagnostics (ID) design influence on overall maintainability attributes, mission readiness, maintenance concept design, and associated Life Cycle Costs (LCC) support concepts.
    i. Unanticipated complex software integration issues affecting all aspects of RAMT.
    j. Lack of adequate ID maturation efforts during system integration.
    k. Failure to anticipate design integration problems where COTS and/or increment design approaches influence RAMT performance.

3. RAMT Techniques and Methodologies

The Enterprise uses the techniques outlined in Figure 1 to aid in RAMT predictions and determining inherent reliability for its systems. During early life cycle phases, the The Enterprise Engineering and Systems Engineering organizations use appropriate techniques for their specific tasks. In the later life cycle stages (once a system design has been prototyped or finalized) each Project accomplishes these techniques as required by their Failure Prevention and Review Board and their Engineering Review Board. Descriptions of these suggested methodologies for some of these techniques are provided in this document. Specific applications and uses are to be detailed in each Project RAMT Process document.

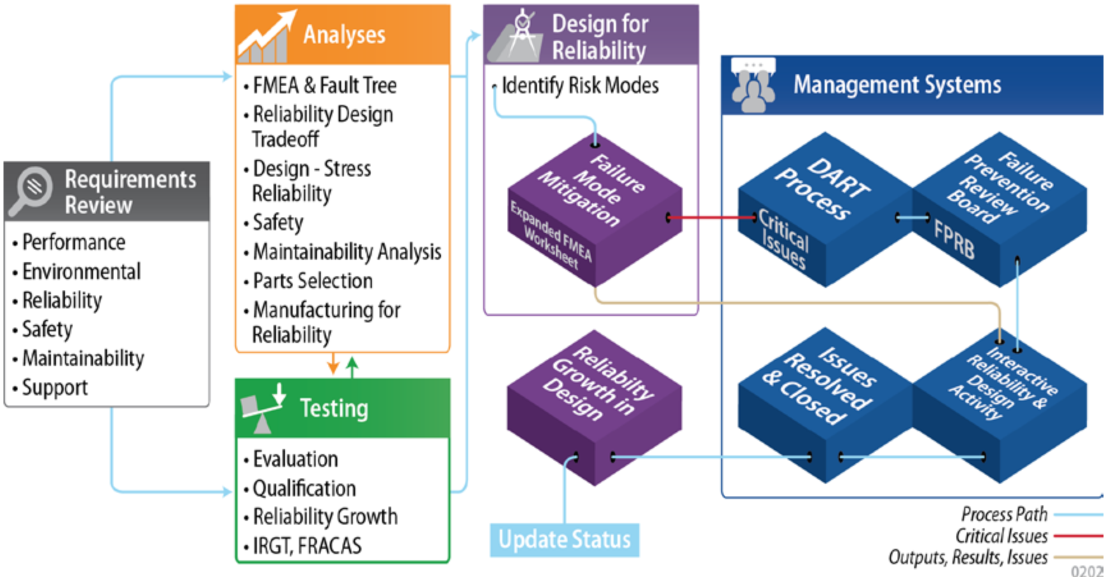


Figure 1: Analysis Techniques

4. Project RAMT Plans, Processes and Procedures

Each Enterprise Project shall develop a specific RAMT Plan (if required), RAMT Process document and RAMT Procedure document for their specific Project, taking into account the requirements of this overall The Enterprise RAMT Plan and existing contractual requirements. A Project is authorized to develop separate Reliability, Availability, Maintainability and Testability processes and procedures if determined more suitable. The Project Systems Engineer and Project Manager must approve of splitting the process and procedure documents.

Each Project Plan, Process and Procedure requires a disciplined application of RAM principles since that is essential to achieving project RAMT goals. This generic The Enterprise RAMT Plan has developed and requires implementation of an effective RAMT program to support achievement of contractually required Project RAMT objectives. A reliability analysis of the requirements is to be conducted in the early stages prior to design. This RAMT Plan is an iterative document and will be maintained throughout the life of each The Enterprise project. It is the implementation plan for all RAMT activities and:

1. Provides visibility into the management and organizational structure of those responsible and accountable for the conduct of RAMT activities over each Project’s life cycle.
2. Defines key resources required to fully implement the RAMT program.
3. Includes a coordinated schedule for conducting all RAMT activities throughout the project life-cycle.
4. Includes detailed descriptions of all RAMT activities, functions, documentation, processes, and strategies required to ensure system RAMT maturation and management throughout the project life cycle.
5. Documents the procedures for verifying that planned activities are implemented and for both reviewing and comparing their status and outcomes.

6. Manages potential RAMT risks due, for example, to new technologies or testing methodology.
7. Flows RAMT allocations and appropriate inputs (e.g., operational and environmental loads) down to suppliers.
8. Includes contingency planning criteria and decision making for altering plans and intensifying RAMT improvement efforts.

5. Reliability Design, Analysis and Predictions

5.1 Skills Required for Reliability Engineering

Each Project must develop a comprehensive process for designing, developing and manufacturing for RAMT that includes people, reporting responsibility, and a RAMT Manager. The following are to be specified in each Project RAMT Process (depending on the system life cycle phase and contract requirements):

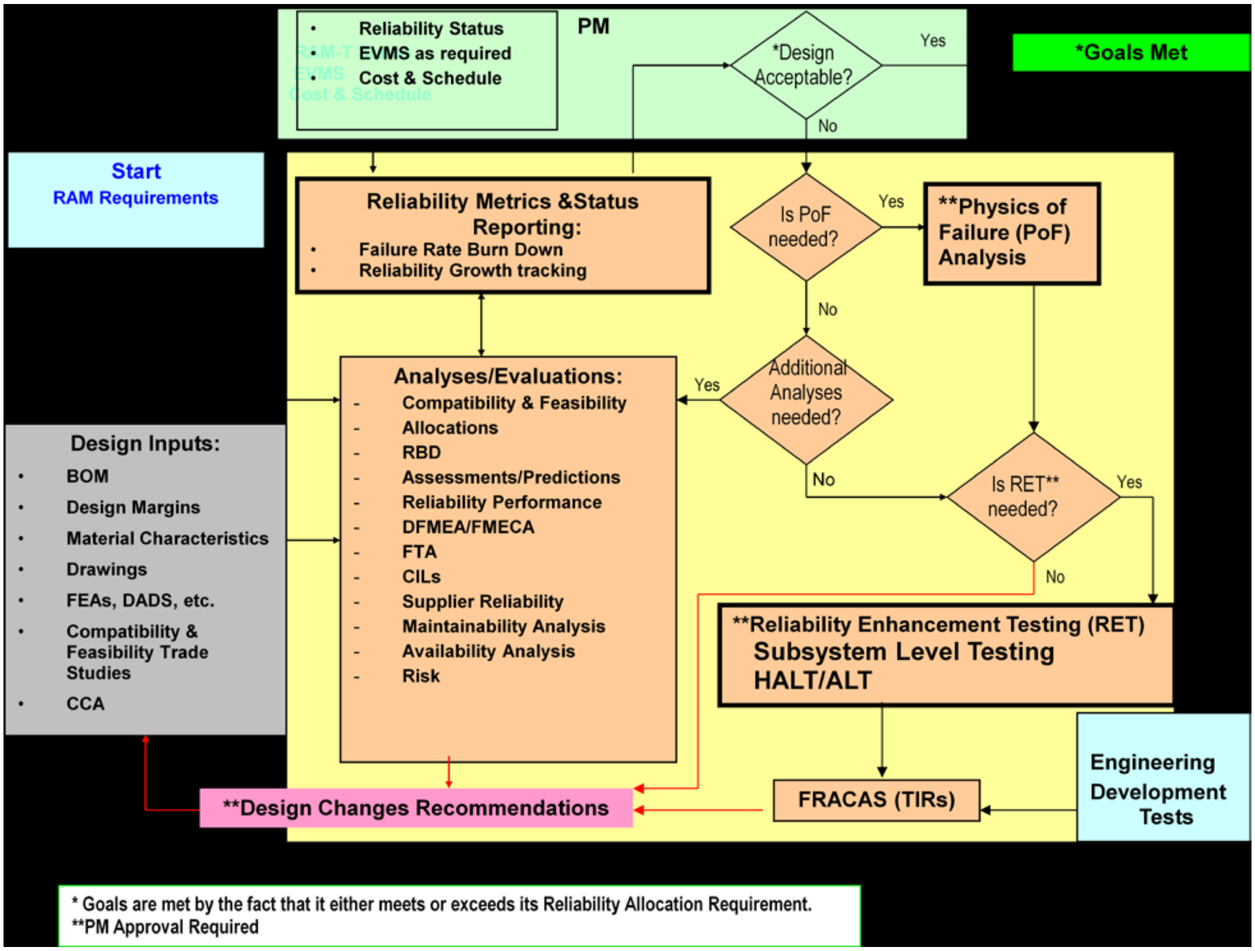
1. Develop a conceptual system model, which consists of components, subsystems, manufacturing processes and performance requirements. Use the model throughout development to estimate performance and RAMT metrics.
2. Identify all critical failure modes and degradations and address them in design.
3. Use data from component-level testing to characterize distribution of times to failure.
4. Conduct sufficient analysis to determine if the design is capable of meeting RAMT requirements.
5. Design in: diagnostics for fault detection, isolation and elimination of false alarms; redundant or degraded system management for enhanced mission success; modularity to facilitate remove-and-replace maintenance; accessibility; and other solutions to user-related needs such as embedded instrumentation and prognostics.

The primary skills that are required to accomplish increased inherent and actual reliability are the ability to understand and anticipate the possible causes of failures, and knowledge of how to prevent them. It is also necessary to have knowledge of the methods that can be used for analyzing designs and data. Effective reliability engineering requires understanding of the basics of failure mechanisms for which experience, broad engineering skills and good knowledge from many different special fields of engineering:

1. Tribology
2. Stress (mechanics)
3. Fracture mechanics/Fatigue (material)
4. Thermal engineering
5. Fluid mechanics/shock loading engineering
6. Electrical engineering
7. Chemical engineering (e.g. Corrosion)
8. Material science

5.2 Reliability Design, Analysis and Predictions

Using industry and DoD best practices and historical experience the Enterprise has identified reliability techniques and methodologies to be used at each stage of a system's life cycle. Note that individual Project Reliability Process documents will define which of these techniques and methodologies are to be used in that specific Project. Each Project shall use a Gap Analysis as a basis for developing their Reliability Process and Procedures. Each system project normally has unique requirements and constraints. Some have considerable new technology or new applications of existing technology. Some consist of minor changes to current system designs. In addition, Enterprise organizations have varying abilities to implement reliability tasks and analyses. A Gap Analysis is a thorough review of project requirements and reliability task capabilities to identify the “gaps”, and to support identification of the “vital few” reliability tasks and methods that will accomplish project objectives within schedule and budget constraints. **The Project Reliability Process documentation will also clearly define the roles and responsibilities that relate to reliability for each project team member and these will be agreed upon and implemented.**



PM
• Reliability Status
• EVMS as required
• Cost & Schedule
*Design Acceptable?
Yes
No
*Goals Met
Start
RAM Requirements
Reliability Metrics &Status Reporting:
• Failure Rate Burn Down
• Reliability Growth tracking
Is PoF needed?
Yes
No
**Physics of Failure (PoF) Analysis
Additional Analyses needed?
Yes
No
Analyses/Evaluations:
- Compatibility & Feasibility
- Allocations
- RBD
- Assessments/Predictions
- Reliability Performance
- DFMEA/FMECA
- FTA
- CILs
- Supplier Reliability
- Maintainability Analysis
- Availability Analysis
- Risk
Design Inputs:
• BOM
• Design Margins
• Material Characteristics
• Drawings
• FEAs, DADS, etc.
• Compatibility & Feasibility Trade Studies
• CCA
Is RET** needed?
Yes
No
**Reliability Enhancement Testing (RET)
Subsystem Level Testing
HALT/ALT
Engineering Development Tests
FRACAS (TIRs)
**Design Changes Recommendations
* Goals are met by the fact that it either meets or exceeds its Reliability Allocation Requirement.
**PM Approval Required

Figure 2: The Enterprise’s Closed Loop Reliability Management Process

For the Concept Stage

During the concept stage it is important to develop and agree on Best Practice Reliability Requirements and get them properly incorporated into technical specifications and flowed down (allocated) to subsystems and components. Properly specifying the reliability at system and component levels can drive the right tasks, both internally and externally with suppliers, in order to achieve high system reliability. Concept stage best practice tasks include:

1. Generate system conditions of use and operating profiles.
2. Integrate reliability into the design process.
3. Develop system level reliability requirements.
4. Flow down reliability requirements to subsystems and components.
5. Generate a system reliability model (also called Reliability Block Diagram or RBD).
6. Perform System FMEA
7. Identify "reliability critical" components and subsystems.

For the Design Stage (Technology Development)

During the design stage of product development, the vital techniques and methodologies that best support Design for Reliability (DFR) must be implemented. It is usually not possible to focus only on reliability testing as the primary way to achieve reliability objectives. It is important to focus on achieving reliability in design, when there are greater opportunities from a cost and feasibility basis. Many of these methods and tools are variously called Robust Design or Design for Six Sigma. Design Stage best practice reliability tasks include:

1. Perform Design Margin Analysis and Design for Reliability[1].
2. Perform Design FMEAs and Process FMEAs. This is due to timing issues. It is usually too late to start Process FMEAs if you wait until the manufacturing stage.
3. Address root cause of known reliability problems.
4. Develop and use product Design Guides.
5. Incorporate reliability input into Design Reviews.
6. Accomplish reliability trade-off studies.
7. Identify and execute specific Robust Design tasks, such as Design of Experiments (DOE), physics of failure modeling and Highly Accelerated Life Testing (HALT).
8. Perform supplier FMEAs for critical components.

For the Assurance Stage (Engineering and Manufacturing Development)

For the Assurance stage, improving the effectiveness of reliability assurance and testing will ensure The Enterprise products are developed and launched with the highest possible system reliability. Properly analyzing test data will markedly increase the effectiveness of all forms of testing to improve product and process reliability. With Product Development times getting shorter and shorter it is essential to accelerate test methods. Doing this properly will not only yield more effective test results but will also facilitate buy in from customers. Assurance stage best practice reliability tasks include:

1. Develop reliability test methods.
2. Develop accelerated life test methods (where appropriate).
3. Execute reliability test plan.
4. Determine parts and materials reliability.
5. Determine impact of software on reliability.
6. Determine ESOH and human factors impacts on reliability.

[1] Design for Reliability provides the process of assuring that the optimum/desired reliability is designed into the item. This process encompasses multiple tools and practices in order to drive reliability into products.

7. Conduct system reliability growth testing.
8. Verify that suppliers meet supplier reliability requirements.
9. Implement ongoing management reviews to include test failure data.
10. Consider the budget that is available for product development and specifically reliability tasks. This is going to drive decisions by making sure that the most critical items are designed with reliability in mind and tested thoroughly. Do not think that you can test for every possible scenario.
11. Understand the degree of risk that is associated with the project/product. The risk can be safety related, it can come from using new technology or by applying existing technology to new applications. Study the regulatory requirements and understand the risk of not meeting them. Understand the risk that your suppliers are introducing into your project.

For the Manufacturing and Launch Stage (Production and Deployment)

Well done Design for Reliability tasks still need to be supported by manufacturing reliability tasks to ensure that the inherent design reliability is not degraded or unstable. During the manufacturing phase, Reliability tasks should primarily focus on reducing or eliminating reliability problems introduced by the manufacturing process. Manufacturing introduces variations in material, processes, manufacturing sites, human operators, contamination, etc. Manufacturing control strategies include Process Control Plans, Statistical Process Control, Identifying and controlling Key Process Characteristics and Design of Experiments in the manufacturing environment. Manufacturing and Launch Stage best practice reliability tasks include:

1. Complete the Process FMEAs.
2. Determine reliability critical items and life-limited items.
3. Develop and execute Integrated Logistics Support.
4. Develop and execute Manufacturing Control Strategies.
5. Monitor and control of subcontractors, suppliers and vendors.
6. Develop and execute Screening and Monitoring plans.
7. Develop and execute a field test plan.
8. Verify that all reliability requirements are met before product launch.
9. Document Product/Program Lessons Learned (during development, production and field).
10. Understand the degree of risk that is associated with the project/product. Understand the risk that your suppliers and the manufacturing processes are introducing into your product.

5.3 Reliability Analysis and Predictions

Reliability engineering for complex systems requires a different, more elaborate systems approach than for non-complex systems. Reliability engineering for most The Enterprise developed/produced systems involve:

1. System availability and mission readiness analysis and related reliability and maintenance requirement allocation
2. Functional System Failure analysis and derived requirements specification

3. Inherent (system) Design Reliability Analysis and derived requirements specification: for both Hardware and Software design
4. System Diagnostics design
5. Predictive and Preventive maintenance (e.g. Reliability Centered Maintenance)
6. Human Factors/Human Interaction/Human Errors
7. Manufacturing and Assembly induced failures (non 0-hour Quality)
8. Maintenance induced failures
9. Transport induced failures
10. Storage induced failures
11. Use (load) studies, component stress analysis and derived requirements specification
12. Software(systematic) failures
13. Failure / reliability testing
14. Field failure monitoring and corrective actions
15. Spare-parts stocking (Availability control)
16. Technical documentation, caution and warning analysis
17. Data and information acquisition/organization (Creation of a general reliability development Hazard Log and FRACAS system)

Many tasks, techniques and analyses used to develop reliability predictions and actual inherent system reliability are specific to particular applications. For The Enterprise developed/produced systems, these include:

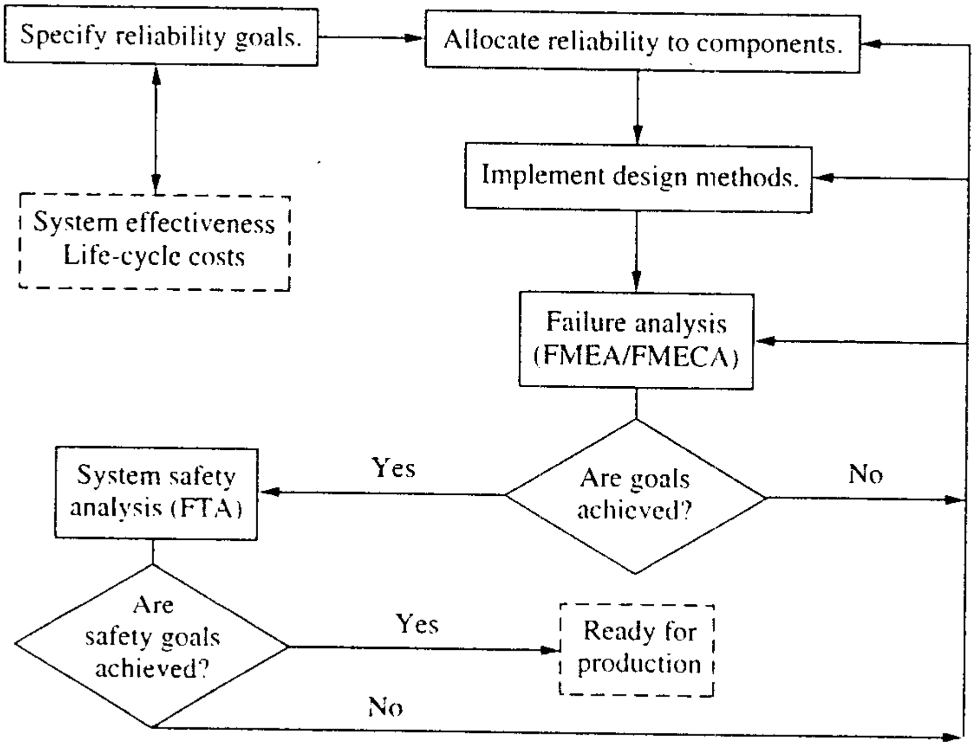
1. Built-in test (BIT) (testability analysis)
2. Failure mode and effects analysis (FMEA) (suggested methodology provided as Appendix A)
3. Reliability hazard analysis
4. Reliability block-diagram analysis
5. Dynamic Reliability block-diagram analysis
6. Fault tree analysis (suggested methodology provided as Appendix B)
7. Root cause analysis (suggested methodology provided as Appendix C)
8. Statistical Engineering, Design of Experiments - e.g. on Simulations/FEM models or with testing
9. Sneak circuit analysis
10. Accelerated testing
11. Reliability growth analysis (reactive reliability)
12. Weibull analysis (for testing or mainly "reactive" reliability)
13. Thermal analysis by finite element analysis (FEA) and/or measurement
14. Thermal induced, shock and vibration fatigue analysis by FEA and/or measurement
15. Electromagnetic analysis
16. Avoidance of single point of failure
17. Functional analysis and functional failure analysis (e.g., function FMEA, FHA or FFA)
18. Predictive and preventive maintenance: reliability centered maintenance (RCM) analysis
19. Testability analysis
20. Failure diagnostics analysis (normally also incorporated in FMEA)

21. Human error analysis
22. Operational hazard analysis
23. Manual screening
24. Integrated Logistics Support Process
25. Environmental Stress Screening

Reliability prediction is the combination of the creation of a proper reliability model of the system under consideration together with estimating (and justifying) the input parameters for this model (like failure rates for a particular failure mode or event and the mean time to repair the system for a particular failure) and finally to provide a system (or part) level estimate for the output reliability parameters (system availability or a particular functional failure frequency). Reliability prediction and actual results are normally to be presented during the system design reviews and logistics reviews. Note that reliability is just one requirement among many system requirements. In The Enterprise Projects, engineering trade studies must be used (and formally documented) to determine the optimum balance between reliability and other requirements and constraints.

5.4 Design for Reliability

Reliability design begins with the development of a system model. Reliability (and availability) models use block diagrams and Fault Tree Analysis to provide a graphical means of evaluating the relationships between different parts of the system. These models may incorporate predictions based on failure rates taken from historical data. While the input data predictions are often not accurate in an absolute sense, they are valuable to assess relative differences in design alternatives. Maintainability parameters, for example MTTR, are other inputs for these models.



Specify reliability goals.
Allocate reliability to components.
System effectiveness
Life-cycle costs
Implement design methods.
Failure analysis (FMEA/FMECA)
Are goals achieved?
Yes
No
System safety analysis (FTA)
Are safety goals achieved?
Yes
Ready for production
No

Figure 3: Reliability Design Process

The most important fundamental initiating causes and failure mechanisms are to be identified and analyzed with engineering tools. A diverse set of practical guidance and practical performance and reliability requirements should be provided to designers so they can generate low-stressed designs and products that protect or are protected against damage and excessive wear. Proper validation of input loads (requirements) may be needed and verification for reliability "performance" by testing may be needed.

Another design technique to prevent failures is called physics of failure. This technique relies on understanding the physical static and dynamic failure mechanisms. It accounts for variation in load, strength and stress leading to failure at high level of detail, possible with use of modern finite element method (FEM) software programs that may handle complex geometries and mechanisms like creep, stress relaxation, fatigue and probabilistic design (Monte Carlo simulations / DOE). The material or component can be re-designed to reduce the probability of failure and to make it more robust against variation. Another common design technique is component derating: Selecting components whose tolerance significantly exceeds the expected stress, such as using a heavier gauge wire that exceeds the normal specification for the expected electrical current.

5.5 Reliability Analysis Techniques

Another effective way to deal with unreliability issues is to perform analysis to be able to predict degradation and being able to prevent unscheduled down events/failures from occurring. RCM (Reliability Centered Maintenance) programs can be used for this. This can be seen in descriptions of events in Fault Tree Analysis, FMEA analysis and a hazard (tracking) log. In this sense language and proper grammar (part of qualitative analysis) plays an important role in reliability engineering, just like it does in safety engineering or in general within systems engineering. Engineers are likely to question why? Such questions are precisely needed because systems engineering is very much about finding the correct words to describe the problem (and related risks) to be solved by the engineering solutions we intend to create. Language in itself is about putting a order in a description of the reality of a (failure of a) complex function/item/ system in a complex surrounding. Reliability engineers use both quantitative and qualitative methods, which extensively use language to pinpoint the risks to be solved.

The importance of language also relates to the risks of human error, which can be seen as the ultimate root cause of almost all failures. As an example, proper instructions (often written by technical authors in so called simplified English) in maintenance manuals, operation manuals, emergency procedures and others are needed to prevent systematic human errors in any maintenance or operational task that may result in system failures.

Software Reliability

Software reliability is defined by the Institute of Electrical and Electronics Engineers (IEEE), much like hardware reliability, as “the probability that software will not cause a system failure for a specified time under specified conditions.” But hardware and software reliability differ in important ways. Hardware failures are generally a result of a combination of a physical fault and a physical or chemical degradation that progresses over time often as a result of stress, shock or other environmental or operating conditions. Software failures are generally caused by inherent faults that were present all along and are discovered during operation when a particular path, system state or loading is experienced. Since software failures are physically different from hardware failures, software failures are often called errors or anomalies, since they generally result from an architectural, logical, or coding error, rather than a physical failure.

Ten guidelines for Software Reliability (and Maintainability) are:

1. Good identification/requirements: Studies have shown the requirements process is the biggest reason for software failures.
2. Modular design: By keeping the lines of code for a particular function packaged together there is less chance of making a software error and less difficulty in troubleshooting one that occurs.
3. Use of high order languages (HOLs): HOLs like C++ are more English-like than assembler language or machine language. Hence, software developers are less likely to make a mistake writing in HOLs.
4. Re-usable software (like pre-packaged, debugged software packages): Like buying a car with a proven engine, re-usable software has less of the "unknown" quality.

5. Use of a single language: Use a single language, if possible, because it does not require translating, converting, or otherwise communicating among several languages, which can be a possible source of error.
6. Fault tolerance: This is the ability to withstand a fault without having an operational failure. It may be achieved by active or inactive redundancy
7. FMEA: If a system includes software, then the FMEA should recognize software as a possible failure point. To neglect the software is to assume it will be error-free.
8. Review and verification via second team: This allows a second independent team to look at the software before it is released.
9. Functional test-debugging the software: Software can be checked on a simulator before it is released.
10. Good documentation: Good documentation will make it easier to trouble-shoot or upgrade software.

Table 1: Reliability Parameter Definitions

Parameter	Description
Failure Rate (λ)	The total number of failures within an item population, divided by the total time expended by that population, during a particular measurement interval under stated conditions.
Hazard Rate	Instantaneous failure rate. At any point in the life of an item, the incremental change in the number of failures per associated incremental change in time.
Mean Time Between Failure (MTBF)	A basic measure of reliability for repairable items. The average time during which all parts of the item perform within their specified limits, during a particular measurement period under stated conditions. (RAC Toolkit)
Mean Time Between Maintenance (MTBM)	A basic measure of reliability for repairable fielded systems. The average time between all system maintenance actions. Maintenance actions may be for repair or preventive purposes. (RAC Toolkit) An alternative definition: The time (i.e. operating hours, flight hours) between the need for maintenance actions to restore a system to fully operational condition, including confirmation that no fault exists (a No Defect maintenance action) This parameter provides the frequency of the need for maintenance and complements the labor hour parameter to project maintenance workload. This parameter is also used to identify unscheduled maintenance (MTBUMA) and Scheduled maintenance (MTBSMA)
Mean Time Between Repair (MTBR)	A basic measure of reliability for repairable fielded systems. The average time between all system maintenance actions requiring removal and replacement or in- situ repairs of a box or subsystem.
Mean Time Between Critical Failure (MTBCF)	A measure of system reliability that includes the effects of any fault tolerance that may exist. The average time between failures that cause a loss of a system function defined as "critical" by the customer. (RAC Toolkit)
Mean Time Between Operational Mission Failure (MTBOMF)	A measure of operational mission reliability for the system. The average time between operational mission failures which cause a loss of the system's "mission" as defined by the customer. This parameter may include both hardware and software "failures."
Mean Time To Failure (MTTF)	A basic measure of reliability for nonrepairable systems. Average failure free operating time, during a particular measurement period under stated conditions.

6. Maintainability Design, Analysis and Prediction

Maintainability is the ability of an item to be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair. Many different parameters are used for maintainability. They include quantitative measures such n time to repair (MTTR), maximum time to repair (M_{Max}), and maintenance ratio (MR). Table 1above lists some of these quantitative measures that are mainly concerned with *time*. Maintainability also is a function of finding failures therefore diagnostics is important and is characterized with metrics such as built-in-test (BIT) effectiveness, fault detection, isolation and false alarm rates. A recent metric that some customers are using is *mean operating hours between false alarms* (MOHBFA). Maintainability is also concerned with economical considerations and ease of maintenance. The ease of maintenance is indirectly indicated, or measured, by:

1. Accessibility,
2. Accuracy of diagnostics,
3. Level of standardization, and
4. Human factors-related considerations.

6.1 Skills Required for Maintainability Engineering

Each Project must develop a comprehensive process for designing, developing and manufacturing for RAMT that includes people, reporting responsibility, and a RAMT Manager. The following are to be specified in each Project RAMT Process (depending on the system life cycle phase and contract requirements):

1. Develop a conceptual system model, which consists of components, subsystems, manufacturing processes and performance requirements. Use the model throughout development to estimate performance and RAMT metrics.
2. Identify all critical failure modes and degradations and address them in design.
3. Use data from component-level testing to characterize distribution of times to failure.
4. Conduct sufficient analysis to determine if the design is capable of meeting RAMT requirements.
5. Design in: diagnostics for fault detection, isolation and elimination of false alarms; redundant or degraded system management for enhanced mission success; modularity to facilitate remove-and-replace maintenance; accessibility; and other solutions to user-related needs such as embedded instrumentation and prognostics.

The primary skills that are required to accomplish increased inherent and actual maintainability are the ability to understand and anticipate the possible causes of required maintenance actions/failures, and knowledge of how to correct/prevent them. It is also necessary to have knowledge of the methods that can be used for analyzing designs and data. Effective maintainability engineering requires understanding of the basics of maintenance required based on a design for which experience, broad engineering skills and good knowledge from many different special fields of engineering:

1. Stress (mechanics)
2. Fracture mechanics/Fatigue (material)
3. Thermal engineering
4. Fluid mechanics/shock loading engineering
5. Electrical engineering
6. Chemical engineering (e.g. Corrosion)
7. Material science

Features of the design (figure 4), such as the level and accuracy of embedded diagnostics instrumentation and prognostics, can increase the maintainability of the system.

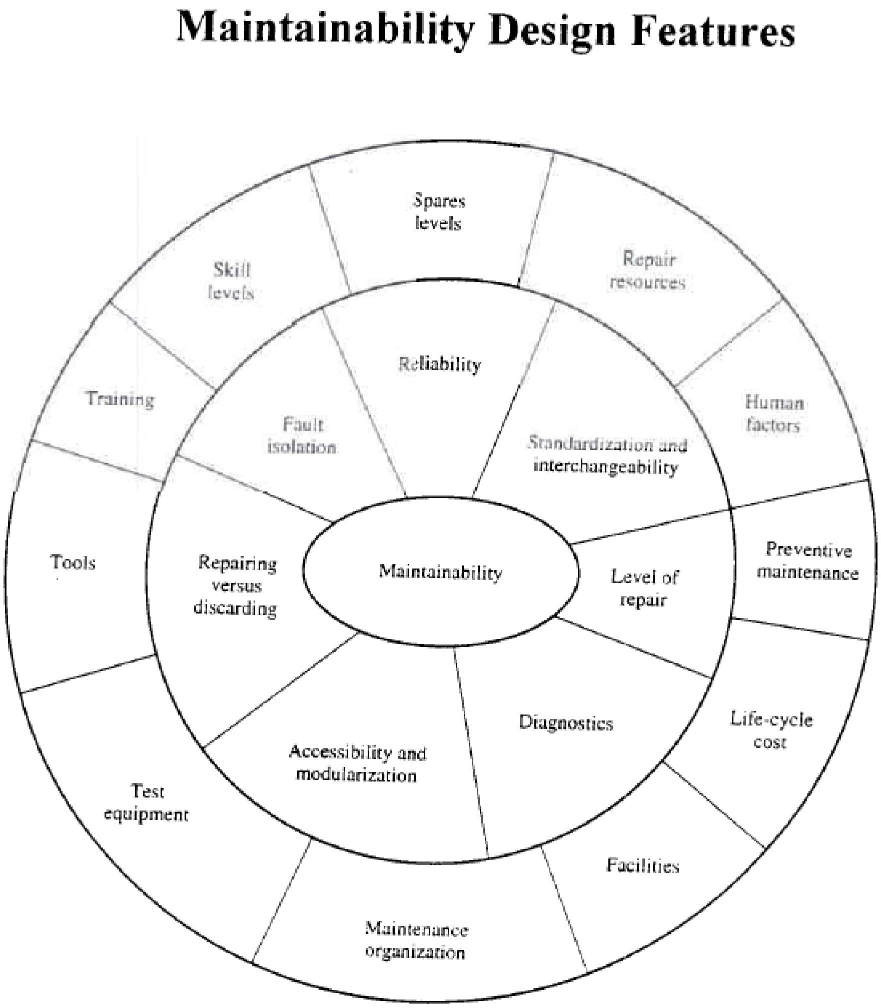


Figure 4: Maintainability Design Features

Some of the more commonly used maintainability metrics are identified in Table 2. Note that testability of a system (see section 8.) is the link between reliability and maintainability. The maintenance strategy can influence the reliability of a system (e.g. by preventive and/or predictive maintenance), although it can never bring it above the inherent reliability. So, Maintainability and Maintenance strategies directly influence the availability of a system. In theory this can be almost unlimited if one would be able to always repair any fault in an infinitely short time. This is in practice impossible. Repairability is always limited due to testability, manpower and logistic considerations.

Table 2: Qualitative Measures of Maintainability

Parameter	Description
Mean Time to Repair (MTTR). Also called Mean Corrective Maintenance Time (M_{ct})	For a sample of repair actions, a composite value representing the arithmetic average of the maintenance cycle times for the individual actions.
Maximum Active Corrective Maintenance Time (M_{max})	That value of maintenance downtime below which one can expect a specified percent of all corrective maintenance actions to be completed. Must be stated at a given percentile point, usually the 90th or 95th. Primarily related to the lognormal distribution.
Mean Preventive Maintenance Time (M_{pt})	A composite value representing the arithmetic average of the maintenance cycle times for the individual preventive maintenance actions (periodic inspection, calibration, scheduled replacement, etc.) for a system.
Median Active Corrective Maintenance Time (M_{ct})	That value of corrective maintenance time that divides all downtime values for corrective maintenance such that 50% are equal to or less than the median and 50% are equal to or greater than the median.
Mean Active Maintenance Time (M_{ct})	The mean or average elapsed time needed to perform maintenance (preventive and corrective), excluding logistic and administrative delays.
Mean Time to Restore System (MTTRS)	For highly redundant systems, the mean or average time needed to switch to a redundant backup unit.
Mean Downtime (MDT)	The mean or average time that a system is not operational due to repair or preventive maintenance. Includes logistics and administrative delays.
Maintenance Labor Hours per Hour or per Cycle, per Action or per time period, e.g. Month	A labor hour factor based on operating or calendar time, maintenance actions, or operating cycles.
Maintenance Ratio (MR)	A measure of the total maintenance labor burden required to maintain an item. It is expressed as the cumulative number of labor hours of maintenance expended in direct labor during a given period divided by the cumulative number of life units during the same period.
Percent BIT Fault Detection (Pfd)	The ratio of the number of faults detected by the system BIT to the total number of faults experienced by the system, expressed as a percent.
Percent BIT Fault Isolation (Pfi)	The ratio of detected faults that was unambiguously isolated to a single replaceable unit or other rule identified in the procurement specification (i.e. to a group of 3 or less replaceable units).
Percent False Alarms (Pfa)	The ratio of detected (indicated) failures to the total indicated failures plus verified failures, expressed as a percent.. For both DT and OT communities, this parameter has now been replaced by MOHBFA
Mean Operating Hours between False Alarm (MOHBFA)	The mean or average time (i.e. operating hours, flight hours) between indicated (detected) faults where no fault could be confirmed. (e.g. False alarm)

7. Availability Design, Analysis and Prediction

Availability is a measure of the degree to which an item is in an operable state and can be committed at the start of a mission when the mission is called for at an unknown (random) point in time and is generally defined as uptime divided by total time (uptime plus downtime). Availability as measured by the user is a function of how often failures occur and corrective maintenance is required, how often preventative maintenance is performed, how quickly indicated failures can be isolated and repaired, how quickly preventive maintenance tasks can be performed, and how long logistics support delays contribute to down time.

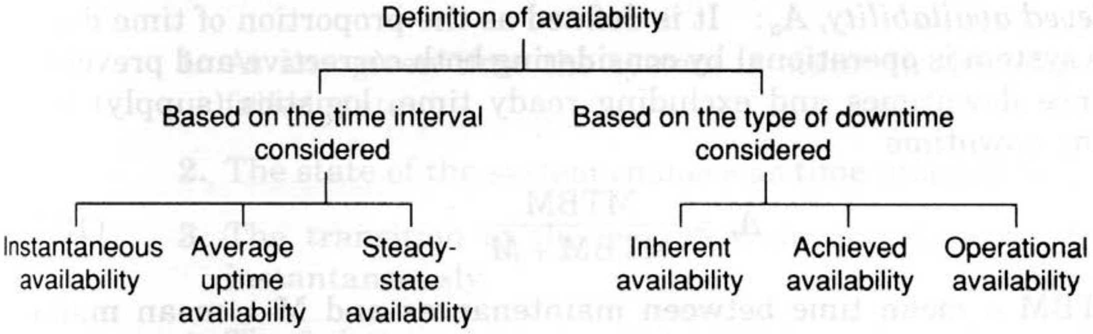

Definition of availability
Based on the time interval considered
Based on the type of downtime considered
Instantaneous availability
Average uptime availability
Steady-state availability
Inherent availability
Achieved availability
Operational availability

Figure 4: Definition of Availability

Each Project must develop a comprehensive process for designing, developing and manufacturing for RAMT that includes people, reporting responsibility, and a RAMT Manager. The following are to be specified in each Project Reliability Process (depending on the system life cycle phase and contract requirements):

1. Develop a conceptual system model, which consists of components, subsystems, manufacturing processes and performance requirements. Use the model throughout development to estimate performance and RAMT metrics.
2. Identify all critical failure modes and degradations and address them in design.
3. Use data from component-level testing to characterize distribution of times to failure.
4. Conduct sufficient analysis to determine if the design is capable of meeting RAMT requirements.
5. Design in: diagnostics for fault detection, isolation and elimination of false alarms; redundant or degraded system management for enhanced mission success; modularity to facilitate remove-and-replace maintenance; accessibility; and other solutions to user-related needs such as embedded instrumentation and prognostics.

The primary skills that are required to accomplish increased inherent and actual availability are the ability to understand and anticipate the possible causes of failures, and knowledge of how to minimize or prevent them. It is also necessary to have knowledge of the methods that can be used for analyzing designs and data. Effective availability engineering requires understanding of the basics of failure mechanisms for which experience, broad engineering skills and good knowledge from many different special fields of engineering:

1. Stress (mechanics)
2. Fracture mechanics/Fatigue (material)

3. Thermal engineering
4. Fluid mechanics/shock loading engineering
5. Electrical engineering
6. Chemical engineering (e.g. Corrosion)
7. Material science

Availability of a system is typically measured as a factor of its reliability - as reliability increases, so does availability. Availability of a system may also be increased by the strategy on focusing on increasing testability and maintainability and not on reliability. Improving maintainability is generally easier than reliability. Maintainability estimates (repair rates) are also generally more accurate. However, because the uncertainties in the reliability estimates are in most cases very large, it is likely to dominate the availability (prediction uncertainty) problem, even while maintainability levels are very high.

When reliability is not under control more complicated issues may arise, like manpower (maintainers / customer service capability) shortage, spare part availability, logistic delays, lack of repair facilities, extensive retro-fit and complex configuration management costs and others. The problem of unreliability may be increased also due to the "domino effect" of maintenance induced failures after repairs. Only focusing on maintainability is therefore not enough. If failures are to be prevented reliability is generally regarded as the most important part of availability.

Reliability needs to be evaluated and improved related to both availability and the total cost of ownership (TCO) (due to cost of spare parts, maintenance man-hours, transport costs, storage cost, part obsolete risks etc.). Often a trade-off is needed between the two. There might be a maximum ratio between availability and TCO. The Project Availability Plan must clearly provide a strategy for availability control. Whether only Availability or also Cost of Ownership is more important depends on the use of the system and contractual requirements. For example, a system that is a critical link in a production system - e.g. a big oil platform – is normally allowed to have a very high cost of ownership if this translates to even a minor increase in availability, as the unavailability of the platform results in a massive loss of revenue which can easily exceed the high cost of ownership. Availability should always be addressed in a Project RAMT analysis in its total context – which includes customer needs. Availability analysis of the system requires knowledge of:

1. How the components are functionally connected.
2. The failure process of the component.
3. The method of operation and the definition of the failure.
4. The repair or maintenance policies

The most simple representation for **availability** is as a ratio of the expected value of the uptime of a system to the aggregate of the expected values of up and down time, or

$$A = \frac{E[\text{Uptime}]}{E[\text{Uptime}] + E[\text{Downtime}]}$$

If we define the status function $X(t)$as

$$X(t) = \begin{cases} 1, & \text{sys functions at time } t \\ 0, & \text{otherwise} \end{cases}$$

Therefore, the availability *A*(*t*) at time *t*>0 is represented by

$$A(t) = \Pr[X(t) = 1] = E[X(t)].$$

Average availability must be defined on an interval of the real line. If we consider an arbitrary constant$c > 0$, then average availability is represented as

$$A_c = \frac{1}{c}\int_0^c A(t)\,dt.$$

Limiting (or steady-state) availability is represented by

$$A = \lim_{c\to\infty} A_c.$$

Limiting average availability is also defined on an interval $[0, c]$as,

$$A_\infty = \lim_{c\to\infty} A_c = \lim_{c\to\infty} \frac{1}{c}\int_0^c A(t)\,dt, \quad c > 0.$$

Example:
If we are using equipment which has a mean time to failure (MTTF) of 81.5 years and mean time to repair (MTTR) of 1 hour:

*MTTF in hours = 81.5*365*24=713940 (This is a reliability parameter and often has a high level of uncertainty!)*

Inherent Availability (Ai) = MTTF/(MTTF+MTTR) = 713940/713941 =99.999859%

Inherent Unavailability = 0.000141%

Outage due to equipment in hours per year = 1/rate = 1/MTTF = 0.01235 hours per year.

Table 3: Category of Elements Determining Availability

Category	Description
Reliability	Mission and non-mission failures that require repair. The lower limit on the number of failures is determined by the inherent level of reliability deigned and built into the system. However, poor manufacturing, inadequate maintenance, operations in conditions beyond those specified for the design, and "acts of God" can increase the number. In addition to determining a lower bound on failures, the reliability characteristics of an item should be considered in determining the number and types of preventive maintenance actions that are either required or are economically desirable.
Maintainability and Maintenance	Maintenance actions include both corrective maintenance (i.e., repairs as a result of failures) and preventive maintenance. The time required for and inherent ease and economy with which a maintenance action can be performed is a direct function of how well maintainability was considered in design. The length of time required for a given maintenance action is also affected by the skill of the maintenance personnel, the maintenance policy and concept, and effectiveness of maintenance manuals and procedures.
Resources	Resources include the number of maintenance personnel available as well as the number and availability of spare and repair parts, support equipment, repair manuals, tools, etc.

8. Testability Design, Analysis and Predictions

Testability addresses the extent to which a system or unit supports fault detection and fault isolation in a confident, timely and cost-effective manner. The incorporation of adequate testability, including built-in test (BIT), requires early end systematic management attention to testability requirements, design and measurement. This document (and MIL-STD-2165) prescribes a uniform approach to The Enterprise Project testability program planning, establishment of testability (including BIT) requirements, testability analysis, prediction, and evaluation, and preparation of testability documentation. Included in any testability program are the following:

1. Testability program planning
2. Testability requirements
3. Testability design
4. Testability prediction
5. Testability demonstration
6. Testability data collect ion end analysis
7. Documentation of testability program
8. Testability review.

To fully understand and determine how to accomplish (based on contractual requirements) each of the above testability requirements, each Project Testability Engineer should read and understand MIL-STD-2165.

8.1 Skills Required for Testability Engineering

Each Project must develop a comprehensive process for designing, developing and manufacturing for RAMT that includes people, reporting responsibility, and a RAMT Manager. The following are to be specified in each Project Testability Process (depending on the system life cycle phase and contract requirements):

1. Develop a conceptual system model, which consists of components, subsystems, manufacturing processes and performance requirements. Use the model throughout development to estimate performance and RAMT metrics.
2. Identify all critical failure modes and degradations and address them in design.
3. Use data from testing to find root cause.
4. Conduct sufficient analysis to determine if the design is capable of meeting RAMT requirements.
5. **Design in:** diagnostics for fault detection, isolation and elimination of false alarms; redundant or degraded system management for enhanced mission success; modularity to facilitate remove-and-replace maintenance; accessibility; and other solutions to user-related needs such as embedded instrumentation and prognostics.
6. **Inference:** Test engineers should be creative, have good intuition, considerable relevant experience and analytical reasoning capability. What can you infer from requirements or specifications?

The primary skills that are required to accomplish increased inherent and actual testability are the ability to understand and anticipate the possible causes of failures, and knowledge of how to test for them. It is also necessary to have knowledge of the methods that can be used for analyzing designs and data for testability. Effective testability engineering requires understanding of the basics of failure mechanisms for which experience, broad engineering skills and good knowledge from many different special fields of engineering:

1. Stress (mechanics)
2. Fracture mechanics/Fatigue (material)
3. Thermal engineering
4. Fluid mechanics/shock loading engineering
5. Electrical engineering
6. Chemical engineering (e.g. Corrosion)
7. Material science

8.2 System Testability

The specific Project Testability Program Plan is the basic tool for establishing and executing an effective testability program. The Project Testability Program Plan shall document what testability tasks are to be accomplished, how each task will be accomplished, when they will be accomplished, and how the results of the task will be used. The Project Testability Program Plan may be a stand-alone document but preferably should be included as part of the systems engineering planning when it is contractually required. The Project Testability Program Plan assist the customer in evaluating the The Enterprise’s approach to and understanding of the testability task requirements, and the organizational structure for performing testability tasks. The Project Testability Program Plan shall be closely coordinated with the Project Maintainability Program Plan.

The selection of tasks which can materially aid the attainment of testability requirements is a difficult problem for both The Enterprise and a specific customer based on contractual funding and schedule constraints. MIL-STD-2165 provides guidance for the selection of tasks based upon identified contractual Project needs. Once appropriate testability program tasks have been selected, each task must be tailored in terms of timing, comprehensiveness and end products to meet the overall Project contractual requirements. Note that any testability program must be an integral part of the systems engineering process and serves as an important link between system design and integrated logistic support

8.3 Software Testability

Software testability is a unique aspect of Testability. It is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context. If the testability of the software artifact is high, then finding faults in the system (if it has any) by means of testing is easier. Testability is not an intrinsic property of a software artifact and cannot be measured directly (such as software size). Instead testability is an extrinsic property which results from interdependency of the software to be tested and the test goals, test methods used, and test resources (i.e., the test context).

A lower degree of testability results in increased test effort. In extreme cases a lack of testability may hinder testing parts of the software or software requirements at all. In order to link the testability with the difficulty to find potential faults in a system (if they exist) by testing it, a relevant measure to assess the testability is how many test cases are needed in each case to form a complete test suite (i.e. a test suite such that, after applying all test cases to the system, collected outputs will let us unambiguously determine whether the system is correct or not according to some specification). If this size is small, then the testability is high. Based on this measure, a testability hierarchy has been proposed.

The testability of software components (modules, classes) is determined by factors such as:

1. Controllability: The degree to which it is possible to control the state of the component under test (CUT) as required for testing.
2. Observability: The degree to which it is possible to observe (intermediate and final) test results.
3. Isolateability: The degree to which the component under test (CUT) can be tested in isolation.
4. Separation of concerns: The degree to which the component under test has a single, well defined responsibility.
5. Understandability: The degree to which the component under test is documented or self-explaining.
6. Automatability: The degree to which it is possible to automate testing of the component under test.
7. Heterogeneity: The degree to which the use of diverse technologies requires to use diverse test methods and tools in parallel.

The testability of software components can be improved by:

1. Test-driven development
2. Design for testability (similar to design for test in the hardware domain)
3. Testability hierarchy

Based on the amount of test cases required to construct a complete test suite in each context (i.e. a test suite such that, if it is applied to the implementation under test, then we collect enough information to precisely determine whether the system is correct or incorrect according to some specification), a testability hierarchy with the following testability classes is to be used:

1. Class I: there exists a finite complete test suite.
2. Class II: any partial distinguishing rate (i.e. any incomplete capability to distinguish correct systems from incorrect systems) can be reached with a finite test suite.
3. Class III: there exists a countable complete test suite.
4. Class IV: there exists a complete test suite.
5. Class V: all cases.

8.4 Requirements Testability

Requirements need to fulfill the following criteria in order to be testable:

1. consistent
2. complete
3. unambiguous
4. quantitative (a requirement like "fast response time" can not be verification/verified)
5. verification/verifiable in practice (a test is feasible not only in theory but also in practice with limited resources)

Treating the requirement as axioms, testability can be treated via asserting existence of a function F_S(software) such that input $I_k$generates output O_k, therefore $F_S : I \to O$. Therefore, the ideal software generates the tuple (I_k, O_k)which is the input-output set Σ, standing for specification.

Now, take a test input I_t, which generates the output O_t, that is the test tuple $\tau = (I_t, O_t)$. Now, the question is whether or not $\tau \in \Sigma$or $\tau \notin \Sigma$. If it is in the set, the test tuple τpasses, else the system fails the test input. Therefore, it is of imperative importance to figure out : can we or can we not create a function that effectively translates into the notion of the set indicator function for the specification set Σ.

By the notion, $1_\Sigma$is the testability function for the specification Σ. The existence should not merely be asserted, should be proven rigorously. Therefore, obviously without algebraic consistency, no such function can be found, and therefore, the specification cease to be termed as testable.

9. Failure Reporting and Corrective Action System

The Failure Reporting and Corrective Action System (FRACAS) is commonly referred to a **"Closed Loop Reporting System"**, and is instrumental in understanding how a system is actually performing in the field from a reliability and maintainability perspective. FRACAS provides a complete reporting, analysis, and corrective action process that fulfills ISO 9000 requirements.

The Enterprise FRACAS objectives include the following:

1. Providing engineering data for corrective action
2. Assessing historical reliability performance, such as Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR), availability, preventive maintenance, etc.
3. Developing patterns for deficiencies
4. Providing data for statistical analysis

5. Aid in measuring contractual performance to better determine warranty information.

9.1 FRACAS Responsibilities

The responsibilities of the keys players would be detailed in the FRACAS procedure and includes the following personnel on all The Enterprise projects:

1. Project Systems Engineer - responsible for ensuring the implementation of corrective actions at the program level as a result of the evaluation and identification of problems relating to R&M issues.
2. Project Manager - responsible for ensuring that project requirements and objectives regarding the failure reporting system are addressed. This includes ensuring that all the responsible parties fulfill their obligations as detailed in the procedure documentation.
3. Reliability and Maintainability Engineer - responsible for reviewing collected "Field Data" for quality and completeness. In addition the engineer will review and analyze all data from all sources to identify any potential problems and trends.
4. System Maintainers - are responsible for the actual collection of "Field Data". The System Maintainer may be the organization implementing the FRACAS or the customer of the system/ product.
5. Production/ QA Staff - Responsible for the collection of failure data during the production phase.
6. Integration & Test Engineering - is responsible for the timely collection of the required R&M "Field Data" as detailed in the procedure during the production phase.
7. Subcontractors - responsible for implementing, as contractually required, a failure reporting system that will address failures for the equipment which they are responsible for. They will, upon completion of their own internal analysis and evaluation, forward failure reports to The Enterprise, who maintains Project FRACAS. These failure reports would provide details of an item's repair to the component level.
8. Suppliers/Vendors - responsible (where required) for providing failure reports for their products to The Enterprise.

9.2 FRACAS Process

A FRACAS could be implemented for a project during the contracted production, integration, test and field deployment phases to provide for the collection and analyses of reliability and maintainability data for the hardware items. These items could be considered to be Line Replaceable Units (LRUs) and Shop Replaceable Assemblies (SRAs).

The FRACAS information would be collected using a report format. Depending upon the complexity of the system or systems and their intended operational profile, this could be achieved by using a simple form or may require use of the TeamCenter FRACAS module. Upon detection of a system failure, which is deemed to be a cause of a hardware failure the FRACAS process would be implemented by initiating the described failure reporting sequence.

1. **Field Data Collection -** The responsible person will collect the required field data. This would vary depending upon project requirements and the phase of the project. This data will be collected to a level commensurate to the level of maintenance performed in the field (e.g. to the LRU or SRA level).

2. **Supplementary Data** - The subcontractors and vendors for units (LRU and SRA) that will be repaired at their facilities are to supply supplementary failure data.
3. Reliability & Maintainability Evaluation - after the collection of the necessary "Field Data" elements the responsible Reliability Engineer will review each Failure Report. This review activity will
    a. Monitor the MTBF data in the case of unwarranted failure patterns and determine possible common failure cause, e.g. SRA, power module yyy
    b. Monitor maintainability characteristics such as BIT effectiveness, accessibility, fault diagnostics problems etc.
    c. Monitor vendor failure report for unwarranted failure trends e.g. same CCA and/ or component, No-Fault-Found (NFF) or No-Evidence-Of-Failure (NEOF), indicates inadequacies in diagnostics capability (BIT, test procedure etc.)
    d. Focus on gray areas such as physical and functional interfaces
    e. Disposition - The responsible Reliability Engineer will disposition a Failure Report with one of the following category types:
    f. No Action Required: Original failure deemed not to of been caused by H/W, H/W failure occurs within equipment which is not considered as part of the primary configuration or no potential problem or unwarranted trend identified
    g. Observation
4. If there are series of failure the Reliability Engineer flags potential problems
5. Recommends/initiates Corrective Actions - The Corrective Action is issued to prevent the reoccurrence of a potential problem. This can be a change in procedures, design recommendations e.g. thermal changes, OTS h/w changes etc.

10. Failure Mode, Effects, and Criticality Analysis

Systems Engineering (RAMT Engineering) shall evaluate the system design to identify potential failure modes and evaluate their effects on subassembly, unit, and system performance. Results shall be incorporated in a Failure Mode, Effects, and Criticality Analysis (FMECA). Preliminary analyses shall be developed during the initial design phase (or as early in the process as possible) using functional block diagrams, boundary diagrams, and parameter or P-diagrams. These preliminary analyses shall be revised and expanded during the detail design phase (any later phase) to identify and analyze failure modes for each functional block and their associated components and subassemblies.

In preparing the FMECA, the following questions shall be addressed:

1. How can each part conceivably fail?
2. What mechanisms might produce this mode of failure?
3. What could the effect be if the failure did occur?
4. Is the failure in the safe or unsafe direction?
5. How is the failure detected?
6. What inherent provisions are provided in the design to compensate for the failure?

The FMECA will analyze the design to the component level and will evaluate the effect of the failure at the subassembly, unit, and system level. Each failure mode will be assigned a Category based on the affect the failure will have on system performance:

1. Cat 1 (Catastrophic): A failure which could adversely affect operator or crew safety.
2. Cat 2 (Critical): A failure which results in total loss of vehicle performance.
3. Cat 3 (Marginal): A failure which results in degraded vehicle performance.
4. Cat 4 (Minor): A failure which has no affect on vehicle performance.

A failure rate will be assigned to each failure mode using data from the Reliability Assessment.

The FMECA shall be used to:

1. Assist in selecting design alternatives with high reliability and high safety potential during the early design phases.
2. Ensure that all conceivable failure modes and their effects on operational success of the system have been considered.
3. List potential failures and identify the severity of their effects.
4. Develop early criteria for test planning and requirements for test equipment.
5. Provide historical documentation for future reference to aid in analysis of field failures and consideration of design changes.
6. Provide a basis for maintenance planning.
7. Provide a basis for quantitative reliability and availability analyses.

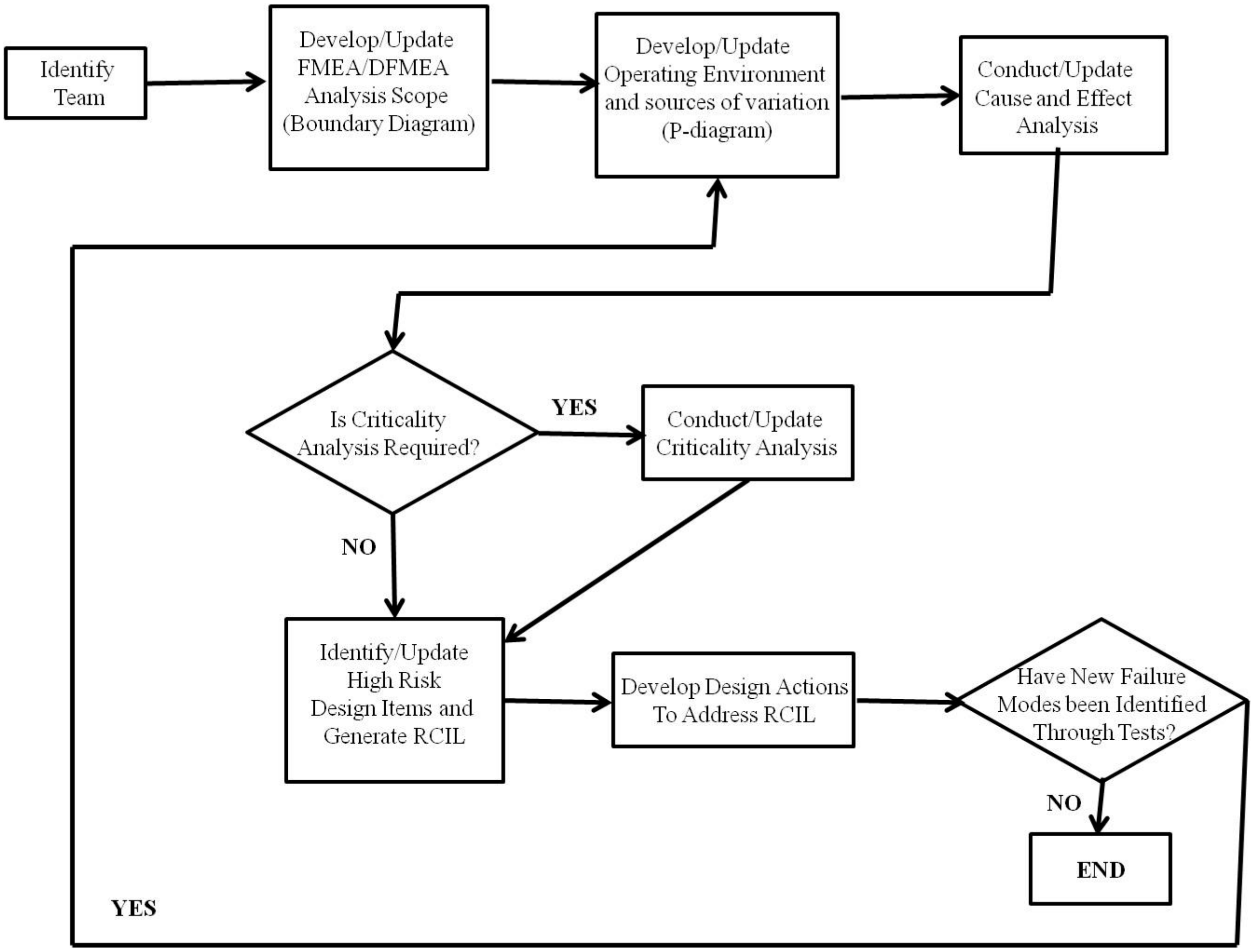

Identify Team
Develop/Update FMEA/DFMEA Analysis Scope (Boundary Diagram)
Develop/Update Operating Environment and sources of variation (P-diagram)
Conduct/Update Cause and Effect Analysis
Is Criticality Analysis Required?
YES
Conduct/Update Criticality Analysis
NO
Identify/Update High Risk Design Items and Generate RCIL
Develop Design Actions To Address RCIL
Have New Failure Modes been Identified Through Tests?
NO
END
YES

Figure 5: DFMEA/FMECA Process

11. Fault Tree Analysis

A Fault Tree Analysis (FTA) analyzes high-level failures and identifies all lower-level (sub-system) failures that cause it. Generally, the undesired event constitutes the highest level (top) event in a fault tree diagram and represents a complete or catastrophic failure of the system. An FTA is also a Risk Management tool that assesses the safety-critical functions within a system's architecture and design. It analyzes high-level failures and identifies all lower-level (sub-system) failures that cause it. FTA is useful during the initial product design phase as a tool for driving the design through an evaluation of both reliability and fault probability perspectives. It can be used to estimate and develop a system's performance reliability requirements to reduce the likelihood of undesired events from occurring.

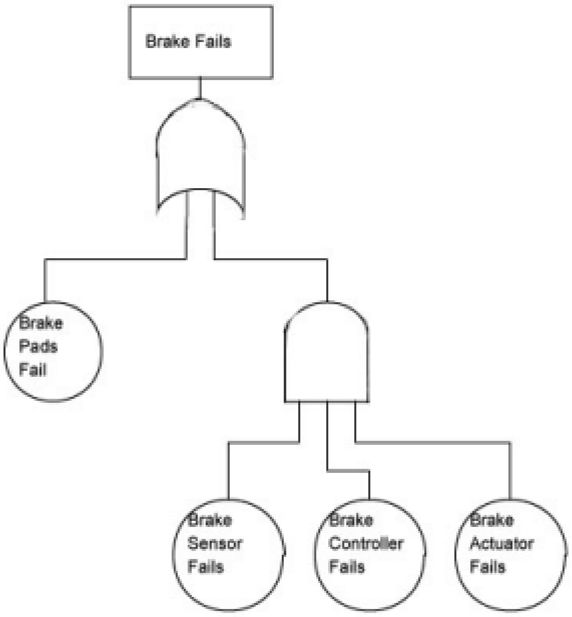


Figure 6: Example Fault Tree Analysis

FTA is particularly useful in functional paths of high complexity in which the outcome of one or more combinations of noncritical events may produce an undesirable critical event. Typical candidates for fault tree analysis are functional paths or interfaces which could have critical impact on system safety, munitions handling safety, safety of operating and maintenance personnel, and probability of error free command in automated systems in which a multiplicity of redundant and overlapping outputs may be involved. The fault tree provides a concise and orderly description of the various combinations of possible occurrences within the system which can result in a predetermined critical output event.

Fault Tree Analysis provides insight into:

1. Functional analysis of highly complex systems
2. Observation of combined effects of simultaneous, non-critical events on the highest level event

3. Evaluation of system reliability
4. Evaluation of human interfaces
5. Evaluations of software interfaces
6. Identification of potential design defects and safety hazards
7. Evaluation of corrective actions
8. Evaluate compliance with the (input) system safety / reliability requirements
   Identification and simplification of maintenance requirements and troubleshooting procedures
9. Elimination of causes for observed failures

Failure Tree Analysis (FTA) is a “top down” method of analysis compared to Failure Modes Effects and Criticality Analysis (FMECA) which is a “bottoms up” method. FTA analysis involves five steps:

1. Define the undesired event to study
2. Obtain an understanding of the system
3. Construct the fault tree
4. Evaluate the fault tree
5. Control the hazards identified

12. Definitions and Acronyms

Anomaly – Any condition that deviates from expectations based on requirements specifications, design documents, user documents, standards, etc. or from someone's perceptions or experiences.

Availability - Availability is a measure of the degree to which an item is in an operable state and can be committed at the start of a mission when the mission is called for at an unknown (random) point in time. Availability as measured by the user is a function of how often failures occur and corrective maintenance is required, how often preventative maintenance is performed, how quickly indicated failures can be isolated and repaired, how quickly preventive maintenance tasks can be performed, and how long logistics support delays contribute to down time. In reliability theory and reliability engineering, the term availability has the following meanings: The degree to which a system, subsystem or equipment is in a specified operable and committable state at the start of a mission, when the mission is called for at an unknown, i.e. a random, time. Simply put, availability is the proportion of time a system is in a functioning condition. This is often described as a mission capable rate. Mathematically, this is expressed as 1 minus unavailability. The ratio of (a) the total time a functional unit is capable of being used during a given interval to (b) the length of the interval. For example, a unit that is capable of being used 100 hours per week (168 hours) would have an availability of 100/168. However, typical availability values are specified in decimal (such as 0.9998). In high availability applications, a metric known as nines, corresponding to the number of nines following the decimal point, is used. With this convention, "five nines" equals 0.99999 (or 99.999%) availability.

Availability, Inherent (Ai) - The probability that an item will operate satisfactorily at a given point in time when used under stated conditions in an ideal support environment. It excludes logistics time, waiting or administrative downtime, and preventive maintenance downtime. It includes corrective maintenance downtime. Inherent availability is generally derived from analysis of an engineering design and is calculated as the mean time to failure (MTTF) divided by the mean time to failure plus the mean time to repair (MTTR). It is based on quantities under control of the designer. Inherent Availability (Ai) = MTTF/(MTTF+MTTR)

Availability, Achieved (Aa) - The probability that an item will operate satisfactorily at a given point in time when used under stated conditions in an ideal support environment (i.e., that personnel, tools, spares, etc. are instantaneously available). It excludes logistics time and waiting or administrative downtime. It includes active preventive and corrective maintenance downtime.

Availability, Operational (Ao) - The probability that an item will operate satisfactorily at a given point in time when used in an actual or realistic operating and support environment. It includes logistics time, ready time, and waiting or administrative downtime, and both preventive and corrective maintenance downtime. This value is equal to the mean time between failure (MTBF) divided by the mean time between failure plus the mean downtime (MDT). This measure extends the definition of availability to elements controlled by the logisticians and mission planners such as quantity and proximity of spares, tools and manpower to the hardware item.

Evaluation - The process of extracting information for decision-makers from test data.

Failure – an event in which an item does not perform one or more of its required functions within the specified limits under specified conditions. A failure can either be catastrophic (total loss of function) or out-of-tolerance (degraded functions beyond specified limits). See Fault.

Failure Symptom – Any circumstance, even t or condition associated with the failure that indicates its existence or occurrence. Failure symptoms can include a temporary intermittent indication that cannot be duplicated.

Failure Effect – The consequence that a particular failure mode has upon the operation, function or status of a product or service.

Failure Mode – The type of defect contributing to a failure, the consequence of the failure (how the failure manifests) or the manner in which the failure is observed.

Failure Mechanism – The process that results in the failure; the process of degradation or chain of events leading to and resulting in a particular failure mode.

Failure Cause – The circumstance that induces or activates a failure mechanism (such as defective soldering, design weakness, assembly techniques, software error, manufacturing process, maintenance error, etc.)

Failure, Relevant – A product or service failure that has been verified and can be expected to occur in normal operational use.

Failure, Non-Relevant – A product or service failure that has been verified as having been caused by a condition not defined for normal operational use.

Failure, Chargeable – A relevant primary failure of the product or service under test and any secondary failure resulting from a single failure incident.
Failure, Non-Chargeable – a non-relevant failure or a relevant failure caused by a previously agreed to set of conditions that eliminates the assignment of failure responsibility to a specific functional group.

Failure, Pattern – The occurrence of two or more failures of the same part or function in identical or equivalent applications where the failures are caused by the same basic failure mechanism and the failure occur at a rate inconsistent with the expected part or function failure rate.

Failure, Multiple – Simultaneous occurrence of two or more verified independent failures. When two or more failed parts are found during troubleshooting and assignable causes cannot be verified as dependent, multiple failures are presumed to have occurred.

Fault – Degradation in performance (as opposed to catastrophic loss) due to failure of parts

Logistics Footprint - The logistics footprint of a system consists of the number of logistics personnel and the materiel needed in a given theater of operations. The ability of a military force to deploy to meet a crisis or move quickly from one area to another is determined in large measure by the amount of logistics assets needed to support that force. Improved RAM reduces the size of the logistics footprint related to the number of required spares, maintenance personnel, and support equipment as well as the force size needed to successfully accomplish a mission.

Maintainability - Maintainability is the ability of an item to be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair. In engineering, maintainability is the ease with which a product can be maintained in order to:

1. isolate defects or their cause,
2. correct defects or their cause,
3. repair or replace faulty or worn-out components without having to replace still working parts,
4. prevent unexpected breakdowns,
5. maximize a product's useful life,
6. maximize efficiency, reliability, and safety,
7. meet new requirements,
8. make future maintenance easier, or
9. cope with a changed environment.

In some cases, maintainability involves a system of continuous improvement - learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience. Note that in telecommunication and several other engineering fields, the term maintainability has the following meanings: A characteristic of design and installation, expressed as the probability that an item will be retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources and the ease with which maintenance of a functional unit can be performed in accordance with prescribed requirements.

Mission Success - Inadequate reliability of equipment directly jeopardizes mission success and may result in undesirable repetition of the mission. The ability to successfully complete a mission is directly affected by the extent to which equipment needed to perform a given mission is available and operating properly when needed. Mission aborts caused by false failure indications can have the same impact as hard failures.

RAM - RAM refers to three related characteristics of a system and its operational support: Reliability, availability, and maintainability.

RAMT – Reliability, Availability, Maintainability and Testability

Readiness - Readiness is the state of preparedness of forces or weapon system or systems to meet a mission, based on adequate and trained personnel, material condition, supplies/reserves of support system and ammunition, numbers of units available, etc. Poor RAM will cause readiness to fall below needed levels or increase the cost of achieving them. Effective diagnostics helps assure both system/mission readiness and efficient repair/return to ready status.

Reliability - Reliability is the probability of an item to perform a required function under stated conditions for a specified period of time. Reliability is further divided into mission reliability and logistics reliability.

Redundancy - One of the most important design techniques. This means that if one part of the system fails, there is an alternate success path, such as a backup system. The reason why this is the ultimate design choice is related to the fact that high confidence reliability evidence for new parts/items is often not available or extremely expensive to obtain. By creating redundancy, together with a high level of failure monitoring and the avoidance of common cause failures, even a system with relative bad single part reliability can be made highly reliable on the system level. Furthermore, by using redundancy and the use of dissimilar design and manufacturing processes (different suppliers) for the single parts/subsystems, less sensitivity for quality issues (early childhood failures) is created and very high levels of reliability can be achieved at all moments of the development cycles (early life times and long term).

System Safety - Inadequate reliability or false failure indications of components deemed Critical Safety Items (CSI) may directly jeopardize the safety of the user(s) of that component's system and result in a loss of life. The ability to safely complete a mission is the direct result of the ability of the CSI associated with the system reliably performing to design intent.

Test - any program or procedure that is designed to obtain, verify, or provide data for the evaluation of any of the following: (1) progress in accomplishing developmental objectives; (2) the performance, operational capability and suitability of systems, subsystems, components, and equipment items; and (3) the vulnerability and lethality of systems, subsystems, components, and equipment items.

Testing - the process of exercising a component or system in order to obtain performance data.

Total Ownership Cost - The concept of Total Ownership Cost (TOC) is an attempt to capture the true cost of design, development, ownership and support of DoD weapons systems. At the individual program level, TOC is synonymous with the life cycle cost of the system. To the extent that new systems can be designed to be more reliable (fewer failures) and more maintainable (fewer resources needed) with no exorbitant increase in the cost of the system or spares, the TOC for these systems will be lower.

13. REFERENCES

1. MIL-HDBK-470A, Department of Defense Handbook: Designing and Developing Maintainable Products and Systems (Volume II), December 1997.

2. MIL STD-785, Reliability Program for Systems and Equipment Development and Production.

3. MIL HDBK-217, Reliability Prediction of Electronic Equipment.

4. MIL STD 756, Reliability Modeling and Prediction.

5. MIL STD-781D, Reliability Testing for Engineering Development, Qualification and Production - Exponential Distribution.

6. MIL STD-470A, Maintainability Program Requirements (for Systems and Equipment).

7. Rodríguez, Ismael; Llana, Luis; Rabanal, Pablo (2014). "A General Testability Theory: Classes, properties, complexity, and testing reductions". IEEE Transactions on Software Engineering 40 (9): 862–894. doi:10.1109/TSE.2014.2331690. ISSN 0098-5589

8. MIL-HDBK-2155, December 1995

9. MIL-STD 471, Maintainability Verification/Demonstration/Evaluation

10. MIL-STD-721, Definition of Effectiveness Terms for Reliability and Maintainability

11. MIL-STD-1309, Definition of Terms for Test, Measurement and Diagnostic Equipment

12. MIL-STD-1388-1, Logistics Support Analysis

13. DOD Ram Guide - DOD Guide for Achieving Reliability, Availability, and Maintainability

14. ANSI-GEIA-STD-0009 - Reliability Program Standard for Systems Design, Development, and Manufacturing

15. IEEE 1332 - Standard Reliability Program for the Development and Production of Electronic Systems and Equipment

16. MIL-HDBK-189 - Reliability Growth Management

17. MIL-STD-810 - Environmental Engineering Considerations and Laboratory Tests

18. MIL-STD-1366 - Transportability Criteria

19. MIL-STD-1472 - Human Engineering Design Criteria

20. MIL-STD-1629 - Procedures for Performing a Failure Mode, Effects, and Criticality Analysis

21. SAE JA 1000 - Reliability Program Standard