Sub-discipline of systems engineering that emphasizes dependability

Reliability engineering is a sub-discipline of systems engineering that emphasizes the ability of equipment to function without failure. Reliability describes the ability of a system or component to function under stated conditions for a specified period of time.[1] Reliability is closely related to availability, which is typically described as the ability of a component or system to function at a specified moment or interval of time.

The reliability function is theoretically defined as the probability of success at time t, which is denoted R(t). This probability is estimated from detailed (physics of failure) analysis, previous data sets or through reliability testing and reliability modelling. Availability, testability, maintainability and maintenance are often defined as a part of "reliability engineering" in reliability programs. Reliability often plays the key role in the cost-effectiveness of systems.

Reliability engineering deals with the prediction, prevention and management of high levels of "lifetime" engineering uncertainty and risks of failure. Although stochastic parameters define and affect reliability, reliability is not only achieved by mathematics and statistics.[2][3] "Nearly all teaching and literature on the subject emphasize these aspects, and ignore the reality that the ranges of uncertainty involved largely invalidate quantitative methods for prediction and measurement."[4] For example, it is easy to represent "probability of failure" as a symbol or value in an equation, but it is almost impossible to predict its true magnitude in practice, which is massively multivariate, so having the equation for reliability does not begin to equal having an accurate predictive measurement of reliability.

Reliability engineering relates closely to Quality Engineering, safety engineering and system safety, in that they use common methods for their analysis and may require input from each other. It can be said that a system must be reliably safe.

Reliability engineering focuses on costs of failure caused by system downtime, cost of spares, repair equipment, personnel, and cost of warranty claims.[5]

History [edit]

The word reliability can be traced back to 1816, and is first attested to the poet Samuel Taylor Coleridge.[6] Before World War II the term was linked mostly to repeatability; a test (in any type of science) was considered "reliable" if the same results would be obtained repeatedly. In the 1920s, product improvement through the use of statistical process control was promoted by Dr. Walter A. Shewhart at Bell Labs,[7] around the time that Waloddi Weibull was working on statistical models for fatigue. The development of reliability engineering was here on a parallel path with quality. The modern use of the word reliability was defined by the U.S. military in the 1940s, characterizing a product that would operate when expected and for a specified period of time.

P

In World War II, many reliability issues were due to the inherent unreliability of electronic equipment available at the time, and to fatigue issues. In 1945, M.A. Miner published the seminal paper titled "Cumulative Damage in Fatigue" in an ASME journal. A main application for reliability engineering in the military was for the vacuum tube as used in radar systems and other electronics, for which reliability proved to be very problematic and costly. The IEEE formed the Reliability Society in 1948. In 1950, the United States Department of Defense formed a group called the "Advisory Group on the Reliability of Electronic Equipment" (AGREE) to investigate reliability methods for military equipment.[8] This group recommended three main ways of working:

Improve component reliability.

Establish quality and reliability requirements for suppliers.

Collect field data and find root causes of failures.

In the 1960s, more emphasis was given to reliability testing on component and system level. The famous military standard MIL-STD-781 was created at that time. Around this period also the much-used predecessor to military handbook 217 was published by RCA and was used for the prediction of failure rates of electronic components. The emphasis on component reliability and empirical research (e.g. Mil Std 217) alone slowly decreased. More pragmatic approaches, as used in the consumer industries, were being used. In the 1980s, televisions were increasingly made up of solid-state semiconductors. Automobiles rapidly increased their use of semiconductors with a variety of microcomputers under the hood and in the dash. Large air conditioning systems developed electronic controllers, as had microwave ovens and a variety of other appliances. Communications systems began to adopt electronics to replace older mechanical switching systems. Bellcore issued the first consumer prediction methodology for telecommunications, and SAE developed a similar document SAE870050 for automotive applications. The nature of predictions evolved during the decade, and it became apparent that die complexity wasn't the only factor that determined failure rates for integrated circuits (ICs). Kam Wong published a paper questioning the bathtub curve[9]â€”see also reliability-centered maintenance. During this decade, the failure rate of many components dropped by a factor of 10. Software became important to the reliability of systems. By the 1990s, the pace of IC development was picking up. Wider use of stand-alone microcomputers was common, and the PC market helped keep IC densities following Moore's law and doubling about every 18 months. Reliability engineering was now changing as it moved towards understanding the physics of failure. Failure rates for components kept dropping, but system-level issues became more prominent. Systems thinking became more and more important. For

software, the CMM model (Capability Maturity Model) was developed, which gave a more qualitative approach to reliability. ISO 9000 added reliability measures as part of the design and development portion of certification. The expansion of the World-Wide Web created new challenges of security and trust. The older problem of too little reliability information available had now been replaced by too much information of questionable value. Consumer reliability problems could now be discussed online in real time using data. New technologies such as micro-electromechanical systems (MEMS), handheld GPS, and hand-held devices that combined cell phones and computers all represent challenges to maintain reliability. Product development time continued to shorten through this decade and what had been done in three years was being done in 18 months. This meant that reliability tools and tasks had to be more closely tied to the development process itself. In many ways, reliability became part of everyday life and consumer expectations.

Overview [edit]

Objective [edit]

The objectives of reliability engineering, in decreasing order of priority, are:[10]

To apply engineering knowledge and specialist techniques to prevent or to reduce the likelihood or frequency of failures. To identify and correct the causes of failures that do occur despite the efforts to prevent them. To determine ways of coping with failures that do occur, if their causes have not been corrected. To apply methods for estimating the likely reliability of new designs, and for analysing 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 products. The primary skills that are required, therefore, 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 analysing designs and data.

Scope and techniques [edit]

Reliability engineering for "complex systems" requires a different, more elaborate systems approach than for non-complex systems. Reliability engineering may in that case involve:

System availability and mission readiness analysis and related reliability and maintenance requirement allocation

Functional system failure analysis and derived requirements specification

Inherent (system) design reliability analysis and derived requirements specification for both hardware and software design

System diagnostics design

Fault tolerant systems (e.g. by redundancy)

Predictive and preventive maintenance (e.g. reliability-centered maintenance)

Human factors / human interaction / human errors

Manufacturing- and assembly-induced failures (effect on the detected "0-hour quality" and reliability)

Maintenance-induced failures

Transport-induced failures

Storage-induced failures

Use (load) studies, component stress analysis, and derived requirements specification

Software (systematic) failures

Failure / reliability testing (and derived requirements)

Field failure monitoring and corrective actions

Spare parts stocking (availability control)

Technical documentation, caution and warning analysis

Data and information acquisition/organisation (creation of a general reliability development hazard log and FRACAS system)

Chaos engineering

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 are required,[11] for example:

Definitions [edit]

Reliability may be defined in the following ways:

The idea that an item is fit for a purpose with respect to time

The capacity of a designed, produced, or maintained item to perform as required over time

The capacity of a population of designed, produced or maintained items to perform as required over time

The resistance to failure of an item over time

The probability of an item to perform a required function under stated conditions for a specified period of time

The durability of an object

Basics of a reliability assessment [edit]

Many engineering techniques are used in reliability risk assessments, such as reliability block diagrams, hazard analysis, failure mode and effects analysis (FMEA),[12] fault tree analysis (FTA), Reliability Centered Maintenance, (probabilistic) load and material stress and wear calculations, (probabilistic) fatigue and creep analysis, human error analysis, manufacturing defect analysis, reliability testing, etc. It is crucial that these analyses are done properly and with much attention to detail to be effective. Because of the large number of reliability techniques, their expense, and the varying degrees of reliability required for different situations, most projects develop a reliability program plan to specify the reliability tasks (statement of work (SoW) requirements) that will be performed for that specific system.

Consistent with the creation of safety cases, for example per ARP4761, the goal of reliability assessments is to provide a robust set of qualitative and quantitative evidence that use of a component or system will not be associated with unacceptable risk. The basic steps to take[13] are to:

Thoroughly identify relevant unreliability "hazards", e.g. potential conditions, events, human errors, failure modes, interactions, failure mechanisms and root causes, by specific analysis or tests.

Assess the associated system risk, by specific analysis or testing.

Propose mitigation, e.g. requirements, design changes, detection logic, maintenance, training, by which the risks may be lowered and controlled for at an acceptable level.

Determine the best mitigation and get agreement on final, acceptable risk levels, possibly based on cost/benefit analysis.

Risk here is the combination of probability and severity of the failure incident (scenario) occurring. The severity can be looked at from a system safety or a system availability point of view. Reliability for safety can be thought of as a very different focus from reliability for system availability. Availability and safety can exist in dynamic tension as keeping a system too available can be unsafe. Forcing an engineering system into a safe state too quickly can force false alarms that impede the availability of the system.

In a de minimis definition, severity of failures includes the cost of spare parts, man-hours, logistics, damage

(secondary failures), and downtime of machines which may cause production loss. A more complete definition of failure also can mean injury, dismemberment, and death of people within the system (witness mine accidents, industrial accidents, space shuttle failures) and the same to innocent bystanders (witness the citizenry of cities like Bhopal, Love Canal, Chernobyl, or Sendai, and other victims of the 2011 TÅ•hoku earthquake and tsunami)â€”in this case, reliability engineering becomes system safety. What is acceptable is determined by the managing authority or customers or the affected communities. Residual risk is the risk that is left over after all reliability activities have finished, and includes the unidentified riskâ€”and is therefore not completely quantifiable.

The complexity of the technical systems such as improvements of design and materials, planned inspections, fool-proof design, and backup redundancy decreases risk and increases the cost. The risk can be decreased to ALARA (as low as reasonably achievable) or ALAPA (as low as practically achievable) levels.

Reliability and availability program plan [edit]

Implementing a reliability program is not simply a software purchase; it is not just a checklist of items that must be completed that will ensure one has reliable products and processes. A reliability program is a complex learning and knowledge-based system unique to one's products and processes. It is supported by leadership, built on the skills that one develops within a team, integrated into business processes and executed by following proven standard work practices.[14]

A reliability program plan is used to document exactly what "best practices" (tasks, methods, tools, analysis, and tests) are required for a particular (sub)system, as well as clarify customer requirements for reliability assessment. For large-scale complex systems, the reliability program plan should be a separate document. Resource determination for manpower and budgets for testing and other tasks is critical for a successful program. In general, the amount of work required for an effective program for complex systems is large.

A reliability program plan is essential for achieving high levels of reliability, testability, maintainability, and the resulting system availability, and is developed early during system development and refined over the system's life-cycle. It specifies not only what the reliability engineer does, but also the tasks performed by other stakeholders. An effective reliability program plan must be approved by top program management, which is responsible for allocation of sufficient resources for its implementation.

A reliability program plan may also be used to evaluate and improve the availability of a system by the strategy of focusing on increasing testability & maintainability and not on reliability. 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, they are likely to dominate the availability calculation (prediction uncertainty problem), even when maintainability levels are very high. When reliability is not under control, more complicated issues may arise, like manpower (maintainers / customer service capability) shortages, 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. Focusing only on maintainability is therefore not enough. If failures are prevented, none of the other issues 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 total cost of ownership (TCO) due to cost of spare parts, maintenance man-hours, transport costs, storage cost, part obsolete risks, etc. But, as GM and Toyota have belatedly discovered, TCO also includes the downstream liability costs when reliability calculations have not sufficiently or accurately addressed customers' personal bodily risks. Often a trade-off is needed between the two. There might be a maximum ratio between availability and cost of ownership. Testability of a system should also be 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.

The reliability plan should 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. For example, a system that is a critical link in a production systemâ€”e.g., a big oil platformâ€”is normally allowed to have a very high cost of ownership if that cost 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. A proper reliability plan should always address RAMT analysis in its total context. RAMT stands for reliability, availability, maintainability/maintenance, and testability in the context of the customer's needs.

Reliability requirements [edit]

For any system, one of the first tasks of reliability engineering is to adequately specify the reliability and maintainability requirements allocated from the overall availability needs and, more importantly, derived from proper

design failure analysis or preliminary prototype test results. Clear requirements (able to designed to) should constrain the designers from designing particular unreliable items / constructions / interfaces / systems. Setting only availability, reliability, testability, or maintainability targets (e.g., max. failure rates) is not appropriate. This is a broad misunderstanding about Reliability Requirements Engineering. Reliability requirements address the system itself, including test and assessment requirements, and associated tasks and documentation. Reliability requirements are included in the appropriate system or subsystem requirements specifications, test plans, and contract statements. Creation of proper lower-level requirements is critical.[15] Provision of only quantitative minimum targets (e.g., Mean Time Between Failure (MTBF) values or failure rates) is not sufficient for different reasons. One reason is that a full validation (related to correctness and verifiability in time) of a quantitative reliability allocation (requirement spec) on lower levels for complex systems can (often) not be made as a consequence of (1) the fact that the requirements are probabilistic, (2) the extremely high level of uncertainties involved for showing compliance with all these probabilistic requirements, and because (3) reliability is a function of time, and accurate estimates of a (probabilistic) reliability number per item are available only very late in the project, sometimes even after many years of in-service use. Compare this problem with the continuous (re-)balancing of, for example, lower-level-system mass requirements in the development of an aircraft, which is already often a big undertaking. Notice that in this case, masses do only differ in terms of only some %, are not a function of time, the data is non-probabilistic and available already in CAD models. In case of reliability, the levels of unreliability (failure rates) may change with factors of decades (multiples of 10) as result of very minor deviations in design, process, or anything else.[16] The information is often not available without huge uncertainties within the development phase. This makes this allocation problem almost impossible to do in a useful, practical, valid manner that does not result in massive over- or under-specification. A pragmatic approach is therefore neededâ€”for example: the use of general levels / classes of quantitative requirements depending only on severity of failure effects. Also, the validation of results is a far more subjective task than for any other type of requirement. (Quantitative) reliability parametersâ€”in terms of MTBFâ€”are by far the most uncertain design parameters in any design.

Furthermore, reliability design requirements should drive a (system or part) design to incorporate features that prevent failures from occurring, or limit consequences from failure in the first place. Not only would it aid in some predictions, this effort would keep from distracting the engineering effort into a kind of accounting work. A design requirement should be precise enough so that a designer can "design to" it and can also proveâ€”through analysis or testingâ€”that the requirement has been achieved, and, if possible, within some a stated confidence. Any type of reliability requirement should be detailed and could be derived from failure analysis (Finite-Element Stress and

Fatigue analysis, Reliability Hazard Analysis, FTA, FMEA, Human Factor Analysis, Functional Hazard Analysis, etc.) or any type of reliability testing. Also, requirements are needed for verification tests (e.g., required overload stresses) and test time needed. To derive these requirements in an effective manner, a systems engineering-based risk assessment and mitigation logic should be used. Robust hazard log systems must be created that contain detailed information on why and how systems could or have failed. Requirements are to be derived and tracked in this way. These practical design requirements shall drive the design and not be used only for verification purposes. These requirements (often design constraints) are in this way derived from failure analysis or preliminary tests. Understanding of this difference compared to only purely quantitative (logistic) requirement specification (e.g., Failure Rate / MTBF target) is paramount in the development of successful (complex) systems.[17]

The maintainability requirements address the costs of repairs as well as repair time. Testability (not to be confused with test requirements) requirements provide the link between reliability and maintainability and should address detectability of failure modes (on a particular system level), isolation levels, and the creation of diagnostics (procedures). As indicated above, reliability engineers should also address requirements for various reliability tasks and documentation during system development, testing, production, and operation. These requirements are generally specified in the contract statement of work and depend on how much leeway the customer wishes to provide to the contractor. Reliability tasks include various analyses, planning, and failure reporting. Task selection depends on the criticality of the system as well as cost. A safety-critical system may require a formal failure reporting and review process throughout development, whereas a non-critical system may rely on final test reports. The most common reliability program tasks are documented in reliability program standards, such as MIL-STD-785 and IEEE 1332. Failure reporting analysis and corrective action systems are a common approach for product/process reliability monitoring.

Reliability culture / human errors / human factors [edit]

In practice, most failures can be traced back to some type of human error, for example in:

Management decisions (e.g. in budgeting, timing, and required tasks)

Systems Engineering: Use studies (load cases)

Systems Engineering: Requirement analysis / setting

Systems Engineering: Configuration control

Assumptions

Calculations / simulations / FEM analysis

Design

Design drawings

Testing (e.g. incorrect load settings or failure measurement)

Statistical analysis

Manufacturing

Quality control

Maintenance

Maintenance manuals

Training

Classifying and ordering of information

Feedback of field information (e.g. incorrect or too vague)

etc.

However, humans are also very good at detecting such failures, correcting for them, and improvising when abnormal

situations occur. Therefore, policies that completely rule out human actions in design and production processes to improve reliability may not be effective. Some tasks are better performed by humans and some are better performed by machines.[18]

Furthermore, human errors in management; the organization of data and information; or the misuse or abuse of items, may also contribute to unreliability. This is the core reason why high levels of reliability for complex systems can only be achieved by following a robust systems engineering process with proper planning and execution of the validation and verification tasks. This also includes careful organization of data and information sharing and creating a "reliability culture", in the same way that having a "safety culture" is paramount in the development of safety critical systems.

Reliability prediction and improvement [edit]

Reliability prediction combines:

creation of a proper reliability model (see further on this page)

estimation (and justification) of input parameters for this model (e.g. failure rates for a particular failure mode or event and the mean time to repair the system for a particular failure)

estimation of output reliability parameters at system or part level (i.e. system availability or frequency of a particular functional failure) The emphasis on quantification and target setting (e.g. MTBF) might imply there is a limit to achievable reliability, however, there is no inherent limit and development of higher reliability does not need to be more costly. In addition, they argue that prediction of reliability from historic data can be very misleading, with comparisons only valid for identical designs, products, manufacturing processes, and maintenance with identical operating loads and usage environments. Even minor changes in any of these could have major effects on reliability. Furthermore, the most unreliable and important items (i.e. the most interesting candidates for a reliability investigation) are most likely to be modified and re-engineered since historical data was gathered, making the standard (re-active or pro-active) statistical methods and processes used in e.g. medical or insurance industries less effective. Another surprising â€“ but logical â€“ argument is that to be able to accurately predict reliability by testing, the exact mechanisms of failure must be known and therefore â€“ in most cases â€“ could be prevented!

Following the incorrect route of trying to quantify and solve a complex reliability engineering problem in terms of MTBF or probability using an-incorrect â€“ for example, the re-active â€“ approach is referred to by Barnard as "Playing the Numbers Game" and is regarded as bad practice.[19]

For existing systems, it is arguable that any attempt by a responsible program to correct the root cause of discovered failures may render the initial MTBF estimate invalid, as new assumptions (themselves subject to high error levels) of the effect of this correction must be made. Another practical issue is the general unavailability of detailed failure data, with those available often featuring inconsistent filtering of failure (feedback) data, and ignoring statistical errors (which are very high for rare events like reliability related failures). Very clear guidelines must be present to count and compare failures related to different type of root-causes (e.g. manufacturing-, maintenance-, transport-, system-induced or inherent design failures). Comparing different types 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 very expensive if done by testing. At the individual part-level, reliability results can often be obtained with comparatively high confidence, as testing of many sample parts might be possible using the available testing budget. However, unfortunately these tests may lack validity at a system-level due to assumptions made at part-level testing. These authors emphasized the importance of initial part- or system-level testing until failure, and to learn from such failures to improve the system or part. The general conclusion is drawn that an accurate and absolute prediction â€“ by either field-data comparison or testing â€“ of reliability is in most cases not possible. An exception might be failures due to wear-out problems such as fatigue failures. In the introduction of MIL-STD-785 it is written that reliability prediction should be used with great caution, if not used solely for comparison in trade-off studies.

Design for reliability [edit]

Design for Reliability (DfR) is a process that encompasses tools and procedures to ensure that a product meets its reliability requirements, under its use environment, for the duration of its lifetime. DfR is implemented in the design stage of a product to proactively improve product reliability.[20] DfR is often used as part of an overall Design for Excellence (DfX) strategy.

Statistics-based approach (i.e. MTBF) [edit]

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 Mean time to repair (MTTR), can also be used as inputs for such models.

The most important fundamental initiating causes and failure mechanisms are to be identified and analyzed with engineering tools. A diverse set of practical guidance as to performance and reliability should be provided to designers so that 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, in addition to verification for reliability "performance" by testing.

A fault tree diagram

One of the most important design techniques is redundancy. 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 or systems is often not available, or is extremely expensive to obtain. By combining redundancy, together with a high level of failure monitoring, and the avoidance of common cause failures; even a system with relatively poor single-channel (part) reliability, can be made highly reliable at a system level (up to mission critical reliability). No testing of reliability has to be required for this. In conjunction with redundancy, the use of dissimilar designs or manufacturing processes (e.g. via different suppliers of similar parts) for single independent channels, can provide less sensitivity to quality issues (e.g. early childhood failures at a single supplier), allowing very-high levels of reliability to be achieved at all moments of the development cycle (from early life to long-term). Redundancy can also be applied in systems engineering by double checking requirements, data, designs, calculations, software, and tests to overcome systematic failures.

Another effective way to deal with reliability issues is to perform analysis that predicts degradation, enabling the prevention of unscheduled downtime events / failures. RCM (Reliability Centered Maintenance) programs can be used for this.

Physics-of-failure-based approach [edit]

For electronic assemblies, there has been an increasing shift towards a different approach 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 that lead to failure with a high level of detail, made possible with the use of modern finite element method (FEM) software programs that can handle complex geometries and mechanisms such as creep, stress relaxation, fatigue, and probabilistic design (Monte Carlo Methods/DOE). The material or component can be re-designed to reduce the probability of failure and to make it more robust against such variations. Another common design technique is component derating: i.e. selecting components whose specifications significantly exceed the expected stress levels, such as using heavier gauge electrical wire than might normally be specified for the expected electric current.

Many of the tasks, techniques, and analyses used in Reliability Engineering are specific to particular industries and applications, but can commonly include:

Results from these methods are presented during reviews of part or system design, and logistics. Reliability is just one requirement among many for a complex part or system. Engineering trade-off studies are used to determine the optimum balance between reliability requirements and other constraints.

The importance of language [edit]

Reliability engineers, whether using quantitative or qualitative methods to describe a failure or hazard, rely on language to pinpoint the risks and enable issues to be solved. The language used must help create an orderly description of the function/item/system and its complex surrounding as it relates to the failure of these functions/items/systems. Systems engineering is very much about finding the correct words to describe the problem (and related risks), so that they can be readily solved via engineering solutions. Jack Ring said that a systems engineer's job is to "language the project." (Ring et al. 2000)[22] For part/system failures, reliability engineers should concentrate more on the "why and how", rather that predicting "when". Understanding "why" a failure has occurred (e.g. due to over-stressed components or manufacturing issues) is far more likely to lead to improvement in the designs and processes used[4] than quantifying "when" a failure is likely to occur (e.g. via determining MTBF). To do this, first the reliability hazards relating to the part/system need to be classified and ordered (based on some form of

qualitative and quantitative logic if possible) to allow for more efficient assessment and eventual improvement. This is partly done in pure language and proposition logic, but also based on experience with similar items. This can for example be seen in descriptions of events in fault tree analysis, FMEA analysis, and hazard (tracking) logs. 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.

Correct use of language can also be key to identifying or reducing the risks of human error, which are often the root cause of many failures. This can include proper instructions in maintenance manuals, operation manuals, emergency procedures, and others to prevent systematic human errors that may result in system failures. These should be written by trained or experienced technical authors using so-called simplified English or Simplified Technical English, where words and structure are specifically chosen and created so as to reduce ambiguity or risk of confusion (e.g. an "replace the old part" could ambiguously refer to a swapping a worn-out part with a non-worn-out part, or replacing a part with one using a more recent and hopefully improved design).

Reliability modeling [edit]

Reliability modeling is the process of predicting or understanding the reliability of a component or system prior to its implementation. Two types of analysis that are often used to model a complete system's availability behavior including effects from logistics issues like spare part provisioning, transport and manpower are fault tree analysis and reliability block diagrams. At a component level, the same types of analyses can be used together with others. The input for the models can come from many sources including testing; prior operational experience; field data; as well as data handbooks from similar or related industries. Regardless of source, all model input data must be used with great caution, as predictions are only valid in cases where the same product was used in the same context. As such, predictions are often only used to help compare alternatives.

A reliability block diagram showing a "1oo3" (1 out of 3) redundant designed subsystem

For part level predictions, two separate fields of investigation are common:

The physics of failure approach uses an understanding of physical failure mechanisms involved, such as mechanical crack propagation or chemical corrosion degradation or failure;

The parts stress modelling approach is an empirical method for prediction based on counting the number and type of components of the system, and the stress they undergo during operation.

Reliability theory [edit]

Reliability is defined as the probability that a device will perform its intended function during a specified period of time under stated conditions. Mathematically, this may be expressed as,

R (t) = P r { T > t } = âˆ« t âˆž f (x) d x {displaystyle R(t)=Pr{T>t}=int _{t}^{infty }f(x),dx !}

where f (x) {displaystyle f(x)!} is the failure probability density function and t {displaystyle t} is the length of the period of time (which is assumed to start from time zero).

There are a few key elements of this definition:

Reliability is predicated on "intended function:" Generally, this is taken to mean operation without failure. However, even if no individual part of the system fails, but the system as a whole does not do what was intended, then it is still charged against the system reliability. The system requirements specification is the criterion against which reliability is measured. Reliability applies to a specified period of time. In practical terms, this means that a system has a specified chance that it will operate without failure before time T {displaystyle T!} Reliability is restricted to operation under stated (or explicitly defined) conditions. This constraint is necessary because it is impossible to design a system for unlimited conditions. A Mars rover will have different specified conditions than a family car. The operating environment must be addressed during design and testing. That same rover may be required to operate in varying conditions requiring additional scrutiny. Two notable references on reliability theory and its mathematical and statistical foundations are Barlow, R. E. and Proschan, F. (1982) and Samaniego, F. J. (2007).

Quantitative system reliability parametersâ€”theory [edit]

Quantitative requirements are specified using reliability parameters. The most common reliability parameter is the mean time to failure (MTTF), which can also be specified as the failure rate (this is expressed as a frequency or conditional probability density function (PDF)) or the number of failures during a given period. These parameters may

be useful for higher system levels and systems that are operated frequently (i.e. vehicles, machinery, and electronic equipment). Reliability increases as the MTTF increases. The MTTF is usually specified in hours, but can also be used with other units of measurement, such as miles or cycles. Using MTTF values on lower system levels can be very misleading, especially if they do not specify the associated Failures Modes and Mechanisms (The F in MTTF).[16]

In other cases, reliability is specified as the probability of mission success. For example, reliability of a scheduled aircraft flight can be specified as a dimensionless probability or a percentage, as often used in system safety engineering.

A special case of mission success is the single-shot device or system. These are devices or systems that remain relatively dormant and only operate once. Examples include automobile airbags, thermal batteries and missiles. Single-shot reliability is specified as a probability of one-time success or is subsumed into a related parameter. Single-shot missile reliability may be specified as a requirement for the probability of a hit. For such systems, the probability of failure on demand (PFD) is the reliability measure â€“ this is actually an "unavailability" number. The PFD is derived from failure rate (a frequency of occurrence) and mission time for non-repairable systems.

For repairable systems, it is obtained from failure rate, mean-time-to-repair (MTTR), and test interval. This measure may not be unique for a given system as this measure depends on the kind of demand. In addition to system level requirements, reliability requirements may be specified for critical subsystems. In most cases, reliability parameters are specified with appropriate statistical confidence intervals.

Reliability testing [edit]

The purpose of reliability testing is to discover potential problems with the design as early as possible and, ultimately, provide confidence that the system meets its reliability requirements.

Reliability testing may be performed at several levels and there are different types of testing. Complex systems may be tested at component, circuit board, unit, assembly, subsystem and system levels.[23] (The test level nomenclature varies among applications.) For example, performing environmental stress screening tests at lower levels, such as piece parts or small assemblies, catches problems before they cause failures at higher levels. Testing proceeds during each level of integration through full-up system testing, developmental testing, and operational testing, thereby

reducing program risk. However, testing does not mitigate unreliability risk.

With each test both a statistical type 1 and type 2 error could be made and depends on sample size, test time, assumptions and the needed discrimination ratio. There is risk of incorrectly accepting a bad design (type 1 error) and the risk of incorrectly rejecting a good design (type 2 error).

It is not always feasible to test all system requirements. Some systems are prohibitively expensive to test; some failure modes may take years to observe; some complex interactions result in a huge number of possible test cases; and some tests require the use of limited test ranges or other resources. In such cases, different approaches to testing can be used, such as (highly) accelerated life testing, design of experiments, and simulations.

The desired level of statistical confidence also plays a role in reliability testing. Statistical confidence is increased by increasing either the test time or the number of items tested. Reliability test plans are designed to achieve the specified reliability at the specified confidence level with the minimum number of test units and test time. Different test plans result in different levels of risk to the producer and consumer. The desired reliability, statistical confidence, and risk levels for each side influence the ultimate test plan. The customer and developer should agree in advance on how reliability requirements will be tested.

A key aspect of reliability testing is to define "failure". Although this may seem obvious, there are many situations where it is not clear whether a failure is really the fault of the system. Variations in test conditions, operator differences, weather and unexpected situations create differences between the customer and the system developer. One strategy to address this issue is to use a scoring conference process. A scoring conference includes representatives from the customer, the developer, the test organization, the reliability organization, and sometimes independent observers. The scoring conference process is defined in the statement of work. Each test case is considered by the group and "scored" as a success or failure. This scoring is the official result used by the reliability engineer.

As part of the requirements phase, the reliability engineer develops a test strategy with the customer. The test strategy makes trade-offs between the needs of the reliability organization, which wants as much data as possible, and constraints such as cost, schedule and available resources. Test plans and procedures are developed for each reliability test, and results are documented.

Reliability testing is common in the Photonics industry. Examples of reliability tests of lasers are life test and burn-in. These tests consist of the highly accelerated aging, under controlled conditions, of a group of lasers. The data collected from these life tests are used to predict laser life expectancy under the intended operating characteristics.[24]

Reliability test requirements [edit]

Reliability test requirements can follow from any analysis for which the first estimate of failure probability, failure mode or effect needs to be justified. Evidence can be generated with some level of confidence by testing. With software-based systems, the probability is a mix of software and hardware-based failures. Testing reliability requirements is problematic for several reasons. A single test is in most cases insufficient to generate enough statistical data. Multiple tests or long-duration tests are usually very expensive. Some tests are simply impractical, and environmental conditions can be hard to predict over a systems life-cycle.

Reliability engineering is used to design a realistic and affordable test program that provides empirical evidence that the system meets its reliability requirements. Statistical confidence levels are used to address some of these concerns. A certain parameter is expressed along with a corresponding confidence level: for example, an MTBF of 1000 hours at 90% confidence level. From this specification, the reliability engineer can, for example, design a test with explicit criteria for the number of hours and number of failures until the requirement is met or failed. Different sorts of tests are possible.

The combination of required reliability level and required confidence level greatly affects the development cost and the risk to both the customer and producer. Care is needed to select the best combination of requirementsâ€”e.g. cost-effectiveness. Reliability testing may be performed at various levels, such as component, subsystem and system. Also, many factors must be addressed during testing and operation, such as extreme temperature and humidity, shock, vibration, or other environmental factors (like loss of signal, cooling or power; or other catastrophes such as fire, floods, excessive heat, physical or security violations or other myriad forms of damage or degradation). For systems that must last many years, accelerated life tests may be needed.

Accelerated testing [edit]

210.00155555556

The purpose of accelerated life testing (ALT test) is to induce field failure in the laboratory at a much faster rate by providing a harsher, but nonetheless representative, environment. In such a test, the product is expected to fail in the lab just as it would have failed in the fieldâ€”but in much less time. The main objective of an accelerated test is either of the following:

To discover failure modes

To predict the normal field life from the high stress lab life

An Accelerated testing program can be broken down into the following steps:

Define objective and scope of the test

Collect required information about the product

Identify the stress(es)

Determine level of stress(es)

Conduct the accelerated test and analyze the collected data.

Common ways to determine a life stress relationship are:

Arrhenius model

Eyring model

Inverse power law model

Temperatureâ€“humidity model

Temperature non-thermal model

Software reliability [edit]

Software reliability is a special aspect of reliability engineering. It focuses on foundations and techniques to make software more reliable, i.e., resilient to faults. System reliability, by definition, includes all parts of the system, including hardware, software, supporting infrastructure (including critical external interfaces), operators and procedures. Traditionally, reliability engineering focuses on critical hardware parts of the system. Since the widespread use of digital integrated circuit technology, software has become an increasingly critical part of most electronics and, hence, nearly all present day systems. Therefore software reliability has gained prominence within the field of system reliability.

There are significant differences, however, in how software and hardware behave. Most hardware unreliability is the result of a component or material failure that results in the system not performing its intended function. Repairing or replacing the hardware component restores the system to its original operating state. However, software does not fail in the same sense that hardware fails. Instead, software unreliability is the result of unanticipated results of software operations. Even relatively small software programs can have astronomically large combinations of inputs and states that are infeasible to exhaustively test. Restoring software to its original state only works until the same combination of inputs and states results in the same unintended result. Software reliability engineering must take this into account.

Despite this difference in the source of failure between software and hardware, several software reliability models based on statistics have been proposed to quantify what we experience with software: the longer software is run, the higher the probability that it will eventually be used in an untested manner and exhibit a latent defect that results in a failure (Shooman 1987), (Musa 2005), (Denney 2005).

As with hardware, software reliability depends on good requirements, design and implementation. Software reliability engineering relies heavily on a disciplined software engineering process to anticipate and design against unintended consequences. There is more overlap between software quality engineering and software reliability engineering than between hardware quality and reliability. A good software development plan is a key aspect of the software reliability program. The software development plan describes the design and coding standards, peer reviews, unit tests,

configuration management, software metrics and software models to be used during software development.

A common reliability metric is the number of software faults per line of code (FLOC), usually expressed as faults per thousand lines of code. This metric, along with software execution time, is key to most software reliability models and estimates. The theory is that the software reliability increases as the number of faults (or fault density) decreases. Establishing a direct connection between fault density and mean-time-between-failure is difficult, however, because of the way software faults are distributed in the code, their severity, and the probability of the combination of inputs necessary to encounter the fault. Nevertheless, fault density serves as a useful indicator for the reliability engineer. Other software metrics, such as complexity, are also used. This metric remains controversial, since changes in software development and verification practices can have dramatic impact on overall defect rates.

Software testing is an important aspect of software reliability. Even the best software development process results in some software faults that are nearly undetectable until tested. Software is tested at several levels, starting with individual units, through integration and full-up system testing. All phases of testing, software faults are discovered, corrected, and re-tested. Reliability estimates are updated based on the fault density and other metrics. At a system level, mean-time-between-failure data can be collected and used to estimate reliability. Unlike hardware, performing exactly the same test on exactly the same software configuration does not provide increased statistical confidence. Instead, software reliability uses different metrics, such as code coverage.

The Software Engineering Institute's capability maturity model is a common means of assessing the overall software development process for reliability and quality purposes.

Structural reliability [edit]

Structural reliability or the reliability of structures is the application of reliability theory to the behavior of structures. It is used in both the design and maintenance of different types of structures including concrete and steel structures.[25][26] In structural reliability studies both loads and resistances are modeled as probabilistic variables. Using this approach the probability of failure of a structure is calculated.

Comparison to safety engineering [edit]

Reliability for safety and reliability for availability are often closely related. Lost availability of an engineering system can cost money. If a subway system is unavailable the subway operator will lose money for each hour the system is down. The subway operator will lose more money if safety is compromised. The definition of reliability is tied to a probability of not encountering a failure. A failure can cause loss of safety, loss of availability or both. It is undesirable to lose safety or availability in a critical system.

Reliability engineering is concerned with overall minimisation of failures that could lead to financial losses for the responsible entity, whereas safety engineering focuses on minimising a specific set of failure types that in general could lead to loss of life, injury or damage to equipment.

Reliability hazards could transform into incidents leading to a loss of revenue for the company or the customer, for example due to direct and indirect costs associated with: loss of production due to system unavailability; unexpected high or low demands for spares; repair costs; man-hours; re-designs or interruptions to normal production.[27]

Safety engineering is often highly specific, relating only to certain tightly regulated industries, applications, or areas. It primarily focuses on system safety hazards that could lead to severe accidents including: loss of life; destruction of equipment; or environmental damage. As such, the related system functional reliability requirements are often extremely high. Although it deals with unwanted failures in the same sense as reliability engineering, it, however, has less of a focus on direct costs, and is not concerned with post-failure repair actions. Another difference is the level of impact of failures on society, leading to a tendency for strict control by governments or regulatory bodies (e.g. nuclear, aerospace, defense, rail and oil industries).[27]

Fault tolerance [edit]

Safety can be increased using a 2oo2 cross checked redundant system. Availability can be increased by using "1oo2" (1 out of 2) redundancy at a part or system level. If both redundant elements disagree the more permissive element will maximize availability. A 1oo2 system should never be relied on for safety. Fault-tolerant systems often rely on additional redundancy (e.g. 2oo3 voting logic) where multiple redundant elements must agree on a potentially unsafe action before it is performed. This increases both availability and safety at a system level. This is common practice in Aerospace systems that need continued availability and do not have a fail-safe mode. For example, aircraft may use triple modular redundancy for flight computers and control surfaces (including occasionally different modes of

operation e.g. electrical/mechanical/hydraulic) as these need to always be operational, due to the fact that there are no "safe" default positions for control surfaces such as rudders or ailerons when the aircraft is flying.

Basic reliability and mission reliability [edit]

The above example of a 2oo3 fault tolerant system increases both mission reliability as well as safety. However, the "basic" reliability of the system will in this case still be lower than a non-redundant (1oo1) or 2oo2 system. Basic reliability engineering covers all failures, including those that might not result in system failure, but do result in additional cost due to: maintenance repair actions; logistics; spare parts etc. For example, replacement or repair of 1 faulty channel in a 2oo3 voting system, (the system is still operating, although with one failed channel it has actually become a 2oo2 system) is contributing to basic unreliability but not mission unreliability. As an example, the failure of the tail-light of an aircraft will not prevent the plane from flying (and so is not considered a mission failure), but it does need to be remedied (with a related cost, and so does contribute to the basic unreliability levels).

Detectability and common cause failures [edit]

When using fault tolerant (redundant) systems or systems that are equipped with protection functions, detectability of failures and avoidance of common cause failures becomes paramount for safe functioning and/or mission reliability.

Reliability versus quality (Six Sigma) [edit]

Quality often focuses on manufacturing defects during the warranty phase. Reliability looks at the failure intensity over the whole life of a product or engineering system from commissioning to decommissioning. Six Sigma has its roots in statistical control in quality of manufacturing. Reliability engineering is a specialty part of systems engineering. The systems engineering process is a discovery process that is often unlike a manufacturing process. A manufacturing process is often focused on repetitive activities that achieve high quality outputs with minimum cost and time.[28]

The everyday usage term "quality of a product" is loosely taken to mean its inherent degree of excellence. In industry, a more precise definition of quality as "conformance to requirements or specifications at the start of use"

is used. Assuming the final product specification adequately captures the original requirements and customer/system needs, the quality level can be measured as the fraction of product units shipped that meet specifications.[29] Manufactured goods quality often focuses on the number of warranty claims during the warranty period.

Quality is a snapshot at the start of life through the warranty period and is related to the control of lower-level product specifications. This includes time-zero defects i.e. where manufacturing mistakes escaped final Quality Control. In theory the quality level might be described by a single fraction of defective products. Reliability, as a part of systems engineering, acts as more of an ongoing assessment of failure rates over many years. Theoretically, all items will fail over an infinite period of time.[30] Defects that appear over time are referred to as reliability fallout. To describe reliability fallout a probability model that describes the fraction fallout over time is needed. This is known as the life distribution model.[29] Some of these reliability issues may be due to inherent design issues, which may exist even though the product conforms to specifications. Even items that are produced perfectly will fail over time due to one or more failure mechanisms (e.g. due to human error or mechanical, electrical, and chemical factors). These reliability issues can also be influenced by acceptable levels of variation during initial production.

Quality and reliability are, therefore, related to manufacturing. Reliability is more targeted towards clients who are focused on failures throughout the whole life of the product such as the military, airlines or railroads. Items that do not conform to product specification will generally do worse in terms of reliability (having a lower MTTF), but this does not always have to be the case. The full mathematical quantification (in statistical models) of this combined relation is in general very difficult or even practically impossible. In cases where manufacturing variances can be effectively reduced, six sigma tools have been shown to be useful to find optimal process solutions which can increase quality and reliability. Six Sigma may also help to design products that are more robust to manufacturing induced failures and infant mortality defects in engineering systems and manufactured product.

In contrast with Six Sigma, reliability engineering solutions are generally found by focusing on reliability testing and system design. Solutions are found in different ways, such as by simplifying a system to allow more of the mechanisms of failure involved to be understood; performing detailed calculations of material stress levels allowing suitable safety factors to be determined; finding possible abnormal system load conditions and using this to increase robustness of a design to manufacturing variance related failure mechanisms. Furthermore, reliability engineering uses system-level solutions, like designing redundant and fault-tolerant systems for situations with high availability

needs (see Reliability engineering vs Safety engineering above).

Note: A "defect" in six-sigma/quality literature is not the same as a "failure" (Field failure | e.g. fractured item) in reliability. A six-sigma/quality defect refers generally to non-conformance with a requirement (e.g. basic functionality or a key dimension). Items can, however, fail over time, even if these requirements are all fulfilled. Quality is generally not concerned with asking the crucial question "are the requirements actually correct?", whereas reliability is.

Reliability operational assessment [edit]

Once systems or parts are being produced, reliability engineering attempts to monitor, assess, and correct deficiencies. Monitoring includes electronic and visual surveillance of critical parameters identified during the fault tree analysis design stage. Data collection is highly dependent on the nature of the system. Most large organizations have quality control groups that collect failure data on vehicles, equipment and machinery. Consumer product failures are often tracked by the number of returns. For systems in dormant storage or on standby, it is necessary to establish a formal surveillance program to inspect and test random samples. Any changes to the system, such as field upgrades or recall repairs, require additional reliability testing to ensure the reliability of the modification. Since it is not possible to anticipate all the failure modes of a given system, especially ones with a human element, failures will occur. The reliability program also includes a systematic root cause analysis that identifies the causal relationships involved in the failure such that effective corrective actions may be implemented. When possible, system failures and corrective actions are reported to the reliability engineering organization.

Some of the most common methods to apply to a reliability operational assessment are failure reporting, analysis, and corrective action systems (FRACAS). This systematic approach develops a reliability, safety, and logistics assessment based on failure/incident reporting, management, analysis, and corrective/preventive actions. Organizations today are adopting this method and utilizing commercial systems (such as Web-based FRACAS applications) that enable them to create a failure/incident data repository from which statistics can be derived to view accurate and genuine reliability, safety, and quality metrics.

It is extremely important for an organization to adopt a common FRACAS system for all end items. Also, it should allow test results to be captured in a practical way. Failure to adopt one easy-to-use (in terms of ease of data-entry for

field engineers and repair shop engineers) and easy-to-maintain integrated system is likely to result in a failure of the FRACAS program itself.

Some of the common outputs from a FRACAS system include Field MTBF, MTTR, spares consumption, reliability growth, failure/incidents distribution by type, location, part no., serial no., and symptom.

The use of past data to predict the reliability of new comparable systems/items can be misleading as reliability is a function of the context of use and can be affected by small changes in design/manufacturing.

Reliability organizations [edit]

Systems of any significant complexity are developed by organizations of people, such as a commercial company or a government agency. The reliability engineering organization must be consistent with the company's organizational structure. For small, non-critical systems, reliability engineering may be informal. As complexity grows, the need arises for a formal reliability function. Because reliability is important to the customer, the customer may even specify certain aspects of the reliability organization.

There are several common types of reliability organizations. The project manager or chief engineer may employ one or more reliability engineers directly. In larger organizations, there is usually a product assurance or specialty engineering organization, which may include reliability, maintainability, quality, safety, human factors, logistics, etc. In such case, the reliability engineer reports to the product assurance manager or specialty engineering manager.

In some cases, a company may wish to establish an independent reliability organization. This is desirable to ensure that the system reliability, which is often expensive and time-consuming, is not unduly slighted due to budget and schedule pressures. In such cases, the reliability engineer works for the project day-to-day, but is actually employed and paid by a separate organization within the company.

Because reliability engineering is critical to early system design, it has become common for reliability engineers, however, the organization is structured, to work as part of an integrated product team.

Education [edit]

Some universities offer graduate degrees in reliability engineering. Other reliability professionals typically have a physics degree from a university or college program. Many engineering programs offer reliability courses, and some universities have entire reliability engineering programs. A reliability engineer must be registered as a professional engineer by the state or province by law, but not all reliability professionals are engineers. Reliability engineers are required in systems where public safety is at risk. There are many professional conferences and industry training programs available for reliability engineers. Several professional organizations exist for reliability engineers, including the American Society for Quality Reliability Division (ASQ-RD),[31] the IEEE Reliability Society, the American Society for Quality (ASQ),[32] and the Society of Reliability Engineers (SRE).[33]

A group of engineers have provided a list of useful tools for reliability engineering. These include: PTC Windchill software, RAM Commander software, RelCalc software, Military Handbook 217 (Mil-HDBK-217), 217Plus and the NAVMAT P-4855-1A manual. Analyzing failures and successes coupled with a quality standards process also provides systemized information to making informed engineering designs.[34]

See also [edit]

References [edit]

N. Diaz, R. Pascual, F. Ruggeri, E. LÃ³pez Droguett (2017). "Modeling age replacement policy under multiple time scales and stochastic usage profiles". International Journal of Production Economics. 188: 22â€“28. doi:10.1016/j.ijpe.2017.03.009.

Further reading [edit]

US standards, specifications, and handbooks [edit]

http://standards.sae.org/ja1000/1_199903/ SAE JA1000/1 Reliability Program Standard Implementation Guide

UK standards [edit]

In the UK, there are more up to date standards maintained under the sponsorship of UK MOD as Defence Standards. The

relevant Standards include:

DEF STAN 00-40 Reliability and Maintainability (R&M)

PART 1: Issue 5: Management Responsibilities and Requirements for Programmes and Plans

PART 4: (ARMP-4)Issue 2: Guidance for Writing NATO R&M Requirements Documents

PART 6: Issue 1: IN-SERVICE R & M

PART 7 (ARMP-7) Issue 1: NATO R&M Terminology Applicable to ARMP's

DEF STAN 00-42 RELIABILITY AND MAINTAINABILITY ASSURANCE GUIDES

PART 1: Issue 1: ONE-SHOT DEVICES/SYSTEMS

PART 2: Issue 1: SOFTWARE

PART 3: Issue 2: R&M CASE

PART 4: Issue 1: Testability

PART 5: Issue 1: IN-SERVICE RELIABILITY DEMONSTRATIONS

DEF STAN 00-43 RELIABILITY AND MAINTAINABILITY ASSURANCE ACTIVITY

PART 2: Issue 1: IN-SERVICE MAINTAINABILITY DEMONSTRATIONS

DEF STAN 00-44 RELIABILITY AND MAINTAINABILITY DATA COLLECTION AND CLASSIFICATION

PART 1: Issue 2: MAINTENANCE DATA & DEFECT REPORTING IN THE ROYAL NAVY, THE ARMY AND THE ROYAL AIR FORCE

PART 2: Issue 1: DATA CLASSIFICATION AND INCIDENT SENTENCINGâ€”GENERAL

PART 3: Issue 1: INCIDENT SENTENCINGâ€”SEA

PART 4: Issue 1: INCIDENT SENTENCINGâ€”LAND

DEF STAN 00-45 Issue 1: RELIABILITY CENTERED MAINTENANCE

DEF STAN 00-49 Issue 1: RELIABILITY AND MAINTAINABILITY MOD GUIDE TO TERMINOLOGY DEFINITIONS

These can be obtained from DSTAN. There are also many commercial standards, produced by many organisations including the SAE, MSG, ARP, and IEE.

French standards [edit]

FIDES [1]. The FIDES methodology (UTE-C 80-811) is based on the physics of failures and supported by the analysis of test data, field returns and existing modelling.

UTE-C 80â€“810 or RDF2000 [2]. The RDF2000 methodology is based on the French telecom experience.

Reference

High Risk: A Doctor's Notes on Pregnancy, Birth, and the Unexpected

Mixed Methods Research: A Guide to the Field (Mixed Methods Research Series)