SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL®
400 Commonwealth Drive, Warrendale, PA 15096-0001

AEROSPACE RECOMMENDED PRACTICE

SAE® ARP5580

Issued 2001-07

Recommended Failure Modes and Effects Analysis (FMEA) Practices for Non-Automobile Applications

FOREWORD

Engineers have always had to consider the effects of component failures on the systems and structures that they design. However, formal methodologies for these types of analyses were not developed until the early 1960s when the obvious safety and reliability requirements of the aerospace industry began to demand them (Reference 2.3.1). In the late 1960s several professional societies began to publish procedures for performing a Failure Modes and Effects Analysis (FMEA). One of the earliest of these was the Society of Automotive Engineers' Aerospace Recommended Practice, ARP926, "Fault/Failure Analysis Procedure" (Reference 2.1.1), published in 1967. In 1974 MIL-STD-1629 (ships), "Procedures for Performing a Failure Mode, Effects and Criticality Analysis" (Reference 2.2.2) was published and, through several revisions, established the basic approach for analyzing a system. By the 1980s FMEA had become a standard part of the design process – at least in the aerospace industry. In 1988 Ford Motor Company published "Potential Failure Mode and Effects Analysis In Design (Design FMEA) and For Manufacturing and Assembly Processes (Process FMEA) Instruction Manual" (Reference 2.3.7) which applied the methodology to manufacturing processes as well as to the design of a product. This procedure focused on the particular needs of the automobile industry and, with input from the major American automobile manufacturing companies and their suppliers, evolved into SAE Surface Vehicle Recommended Practice, J1739 (Reference 2.1.4), issued by the SAE in 1994.

In June of 1994, then Secretary of Defense William Perry issued a memorandum titled "Specifications and Standards – A New Way of Doing Business", which directed the Department of Defense to increase their reliance on commercial products and practices. As a result of "the Perry Memo", many US military standardization documents, including MIL-STD-1629, were cited for cancellation. Around the same time the Defense Standards Improvement Council (DSIC) was chartered to oversee the standardization reform process. DSIC coordinated its position on MIL-STD-1629 with the Society of Automotive Engineers, which through its G-11, Reliability, Maintainability, Supportability and Logistics (RMSL) Division had already chartered a subcommittee to create a new FMEA procedure updating MIL-STD-1629. The subcommittee was comprised of representatives from industry, government and academia. In response to that charter, this recommended best practice has been developed.

TO PLACE A DOCUMENT ORDER: (724) 776-4970 FAX: (724) 776-0790 SAE WEB ADDRESS: http://www.sae.org

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

1. SCOPE:

Recommended Failure Modes and Effects Analysis (FMEA) Practices For Non-Automobile Applications describes the basic procedures for performing a Failure Modes and Effects Analysis (FMEA). It encompasses functional, interface, and detailed FMEA, as well as certain pre-analysis activities (FMEA planning and functional requirements analysis), post-analysis activities (failure latency analysis, FMEA verification, and documentation), and applications to hardware, software, and process design. It is intended for use by organizations whose product development processes use FMEA as a tool for assessing the safety and reliability of system elements, or as part of their product improvement processes. A separate, Surface Vehicle Recommended Practice, J1739, is intended for use in automobile applications.

1.1 Purpose:

In developing this procedure the subcommittee has endeavored to develop a procedure that reflects the best current commercial practices. This procedure was developed in recognition of today's intense and competitive market demands for high reliability, affordability, and speed to market. The subcommittee had several objectives in defining the FMEA process:

1. Define a basic methodology to include functional, interface, and detailed FMEA. This will facilitate performing the analysis throughout the design process, from early in the conceptual stage to implementation and production.

2. Extend the methodology to include both product and process FMEAs. The methodology can be applied to the many technologies (e.g., mechanical, electrical, software, etc.) used in the development of a product. This helps to facilitate communications between all the parties involved in the development of a system and is useful in a concurrent engineering environment.

3. Provide simple techniques for ranking failure modes for corrective actions and for identifying fault equivalencies.

4. Define the types of information needed for the FMEA in electronic databases, thus facilitating semi-automation of the analysis.

5. Provide procedures for managing the FMEA and for getting the most benefit from the analysis.

2. REFERENCES:

The following publications are provided for information purposes only and are not a required part of this SAE Technical Report.

2.1 SAE Publications:

Available from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

2.1.1 “Fault/Failure Analysis Procedure”, Society of Automotive Engineers, Aerospace Recommended Practice, ARP926, Sept. 15, 1967, ARP926A, Nov. 15, 1979.

2.1.2 “Fault/Failure Analysis For Digital Systems and Equipment”, Society of Automotive Engineers, Aerospace Recommended Practice, ARP1834, Aug. 1986.

2.1.3 Reliability, Maintainability, and Supportability Guidebook, SAE International RMS Committee (G-11), 2nd Ed. Society of Automotive Engineers, 1992.

2.1.4 “Potential Failure Mode and Effects Analysis In Design (Design FMEA) and Potential Failure Mode and Effects Analysis In Manufacturing and Assembly Processes (Process FMEA) Reference Manual”, Society of Automotive Engineers, Surface Vehicle Recommended Practice, J1739, July 1994.

2.1.5 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment”, Society of Automotive Engineers, Aerospace Recommended Practice, ARP4761, December 1996.

2.2 U.S. Government Publications:

Available from DODSSP, Subscription Services Desk, Building 4D, 700 robbins Avenue, Philadelphia, PA 19111-5094.

2.2.1 “Electronic Reliability Design Handbook”, MIL-HDBK-338-1, Volume I, Oct. 12, 1988.

2.2.2 “Procedures For Performing A Failure Mode Effects and Criticality Analysis”, US MIL-STD-1629 (ships) Nov. 1, 1974; US MIL-STD-1629A, Nov. 24, 1980; US MIL-STD-1629A/Notice 2, Nov. 28, 1984.

2.2.3 “Reliability Prediction of Electronic Equipment”, MIL-HDBK-217F, Dec. 10, 1993.

2.2.4 “System Design and Analysis”, Advisory Circular 25.1309-1A, Federal Aviation Administration (FAA), June 1988.

2.2.5 “Fault Tree Handbook”, NUREG-0492, U.S. Nuclear Regulatory Commission, Jan. 1981.

2.3 Applicable References:

2.3.1 J. S. Coutinho, “Failure-Effect Analysis”, Trans. New York Academy of Sciences, 1964, pp. 564-584.

2.3.2 “Failure Mode, Effects, and Criticality Analysis (FMECA)”, CRTA-FMECA, Reliability Analysis Center, Rome, NY, 1993.

2.3.3 “Nonelectronic Parts Reliability Data -1995”, NPRD-95, Reliability Analysis Center, Rome NY, 1995.

2.3.4 “Failure Mode/Mechanism Distributions 1997”, FMD-97, 1997, Reliability Analysis Center.

2.3.5 “Analysis Techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)”, International Electrotechnical Commission, IEC Standard Pub. 812, 1985.

2.3.6 “Failure Mode and Effect Analyses”, Electronic Industries Association G-41 Committee on Reliability, Reliability Bulletin No. 9, November 1971.

2.3.7 “Potential Failure Mode and Effects Analysis In Design (Design FMEA) and For Manufacturing and Assembly Processes (Process FMEA) Instruction Manual”, Ford Motor Company, Sept 1988.

2.3.8 “Reliability Prediction Procedure for Electronic Equipment”, Bellcore, TR-TSY-332, Issue 5, December 1995.

2.3.9 “Software Considerations in Airborne Systems and Equipment Certification”, Radio Technical Commission for Aeronautics, RTCA/DO-178B, Dec. 1992.

2.3.10 C. S. Spangler, “Systems Engineering – The Fault Analysis Process For Commercial Avionics Application,” Proceedings of the Third Annual International Symposium of the National Council on Systems Engineering, 1993.

2.3.11 C. S. Spangler, “Equivalence Relations within the Failure Mode and Effects Analysis”. Proc. Ann. Reliability and Maintainability Symp. (Washington, DC), 1999, pp. 352-357.

2.3.12 P. L. Goddard, “Validating The Safety Of Embedded Real-Time Control Systems Using FMEA”, Proc. Ann. Reliability and Maintainability Symposium, January 1993, pp. 227-230.

2.3.13 M. A. Friedman, P. Y. Tran, and P. L. Goddard, Reliability of Software Intensive Systems, Noyes Data Corporation, ISBN: 0-8155-1361-5, 1995.

2.3.14 M. A. Friedman, J. Voas, Software Assessment: Reliability, Safety, Testability, John Wiley, ISBN: 0-4710-1009-X, 1995.

2.3.15 R. S. Carson, “A Set Theory Model for Anomaly Handling in System Requirements Analysis”, Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering, 1995.

2.3.16 P. D. T. O’Connor, Practical Reliability Engineering, 3rd revised Ed., John Wiley, 1995.

2.4 Definitions:

ALLOCATION: The results of the process of assigning an identified portion of a functional requirement to a specific item of hardware or software, a facility, or to personnel.

BUILT-IN-TEST (BIT): Diagnostic tests included as part of the system design.

BOTTOM-UP ANALYSIS: Analysis of a component, part or subsystem which starts with the failure modes of the lowest indenture level items of the system and successively iterates through the next higher levels ending at the system level.

2.4 (Continued):

CIRCUIT: A description of the task, action, or operation performed by a group of parts at the lowest indenture level.

COMPENSATING PROVISIONS: Design provisions, or operator actions, which circumvent or mitigate the effects of a failure. Compensating design provisions are features at any indenture level that will nullify the effects of a malfunction or failure but do not prevent its occurrence.

COMPONENT TYPE: Classification of a piece-part based on its characteristics and the ways in which it typically fails. Examples: digital integrated circuits; resistors; capacitors; transformers; valves; actuators; air conditioners; batteries; condensers; compressors; filters; fans; fuses; hoses; springs; regulators; relays; seals; pumps; switches; transistors; etc. Piece-part failure modes are postulated based on the “Component Type”.

COMPUTER SOFTWARE CONFIGURATION ITEM (CSCI): An aggregation of software that satisfies an end-use function and is intended for separate configuration management by the acquirer. CSCIs are selected based on tradeoffs among software function, size, host, or target computers, developer, support concept, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors.

COMPUTER SOFTWARE COMPONENT (CSC): An aggregation of software which is part of a CSCI that satisfies one or more end-use functions. A CSC is generally composed of more than one software unit. CSCs are selected based on tradeoffs among software function, size, host, or target computers, developer, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors.

CORRECTIVE ACTION: A documented design, process, procedure or materials change implemented and validated to eliminate design deficiencies or mitigate failure consequences.

CRITICALITY: A relative measure of the impact of a failure mode on the mission objective. Criticality combines the frequency of occurrence and the level of severity of a failure mode.

CRITICALITY ANALYSIS: A procedure by which each potential failure mode is ranked according to the combined influence of its severity and probability of occurrence.

DETAILED FAILURE MODE & EFFECTS ANALYSIS (DETAILED FMEA): An analysis that assesses the failure causes and effects of failure modes on the individual items (piece-parts, software routines, or process steps) that comprise the system being analyzed.

DETAILED FAILURE MODES, EFFECTS & CRITICALITY ANALYSIS (DETAILED FMECA): An extension of the Detailed FMEA to include an assessment of failure mode severity and probability of occurrence.

DETECTED FAILURE: A postulated failure mode whose presence can be discovered. See Detection Mechanism.

2.4 (Continued):

DETECTION MECHANISM: The means or methods by which a failure can be discovered. Some common detection mechanisms are: (1) an operator under normal system operation; and (2) a maintenance crew by some diagnostic action.

END EFFECT: See End Level Effect.

END-ITEM: The highest level item in a hierarchical analysis of a system. See Item.

END LEVEL EFFECT: The impact or consequence of a failure mode on the operation, function, or status of the end-item. This is derived from analyzing the effects of a failure mode on the major subsystems that make up the complete system. See Mission Impact.

EXPOSURE TIME: The period (in clock time or cycles) during which an item is exposed to a failure. The period is measured from when the function was verified to be functioning to when it is verified again.

FAILURE: The inability of an item to perform its required function within previously specified limits.

FAILURE ANALYSIS: The logical, systematic examination of an item or its diagrams to identify and analyze the probability, causes, and consequences of potential and real failures.

FAILURE CAUSE: The physical or chemical processes, design defects, quality defects, part misapplication, or other processes which are the basic reason for failure, or which initiate the process which leads to failure. Failure cause answers the question “Why does the part fail?”

FAILURE EFFECT: The consequences of a failure mode on the operation, function, or status of an item. Failure effects are classified as local effect, next higher level effect, and end effect.

FAILURE MECHANISM: The process involved in the cause of failure. Failure Mechanism answers the question “What is the failure process?”

FAILURE MODE: The manner in which an item fails. Failure Mode answers the question “How does the part fail?”

FAILURE MODE RATIO: The fraction of item failures apportioned to the failure mode under consideration.

FAILURE MODE AND EFFECTS ANALYSIS (FMEA): A procedure by which each potential failure mode or fault of a system is analyzed to determine the consequences or effects thereof on the system, to classify each potential failure mode according to its severity, and to recommend actions to eliminate, or compensate for, unacceptable effects.

FAILURE MODE, EFFECTS AND CRITICALITY ANALYSIS (FMECA): An extension of the FMEA procedure to include assessment of the failure mode severity and probability of occurrence.

2.4 (Continued):

FAILURE-REPORTING-AND-CORRECTIVE-ACTION-SYSTEM (FRACAS): A system for reporting failures and tracking them to ensure that corrective actions are taken to correct the problem.

FAN-IN: The specified number of standard load sources that drive an input without impairing normal operations.

FAN-OUT: The specified number of standard loads that an output can drive without impairing normal operations.

FAULT: An anomaly in the functional operation of an equipment or system.

FAULT DETECTION: The process of discovering that a failure has occurred.

FAULT EQUIVALENT FAILURE MODES: A set of failure modes whose consequences are identical.

FAULT INDICATION: The means by which a failure is detected: visual, audible, odorous, console display, printout, etc.

FAULT ISOLATION: The process of determining the location of a fault to effect repair.

FAULT SIGNATURE: Indicators by which a specific system fault can be recognized. It has two parts: (1) the test conditions, consisting of the operator or test stimulus; and (2) the primary output response consisting of measurable parameter changes or monitor indications by which the specific system fault can be recognized.

FAULT TREE: A pictorial representation of the combinations of subsystem and component failures which can result in a specific, often hazardous, system event. The fault tree, when annotated with failure probabilities, can be evaluated to establish the probability of the ultimate undesirable end-level event occurring as a function of the probabilities of the identifiable contributory events.

FD/FI: See Fault Detection, Fault Isolation.

FMEA: See Failure Mode and Effects Analysis.

FMECA: See Failure Mode, Effects and Criticality Analysis.

FUNCTION: Actions which must be performed by an element or elements of the system in order for the system to accomplish its intended purpose.

Within the context of Functional Analysis a function is a description of the task, duty, action, or operation performed by a group of elements at the functional building block level. See Functional Block Diagram.

2.4 (Continued):

FUNCTION TYPE: A classification of a functional block (or function) based on the single task, duty, action, or operation it performs. The classification is done irrespective of the physical implementation of the item. Examples: amplifier, filter, mixer, etc. Functional failure modes are postulated on a per “Function Type” basis. See Function, Functional Block Diagram.

FUNCTIONAL ANALYSIS: The process of examining the characteristics of a defined function to identify all necessary sub-functions. The sub-functions are then arranged to show their relationships.

FUNCTIONAL BLOCK DIAGRAM: A diagram which illustrates the operation, interrelationships, and interdependencies of the functions (shown as functional blocks) of a system/subsystem. See Function, Function Type.

FUNCTIONAL FAILURE MODE AND EFFECTS ANALYSIS (FUNCTIONAL FMEA): A top-level analysis of the failure effects of a system (or subsystem) based upon a failure mode occurring within functional partitions of the system (or subsystem).

FUNCTIONAL FAILURE MODE, EFFECTS & CRITICALITY ANALYSIS (FUNCTIONAL FMECA): An extension of the Functional FMEA to include an assessment of the failure mode severity and probability of occurrence.

HARDWARE: A composite, at any level of complexity, of equipment which is designated to perform a specific function or mission.

HIERARCHY: A partitioning scheme that establishes an ordered relationship between the items in a system. A parent/child or “contained in” ordering is most often used to establish a hierarchical system ordering.

HIGHEST LEVEL INDENTURE: The most complex division of the specific item being analyzed. See Indenture Level.

INDENTURE LEVEL: The item levels which identify or describe the relative complexity of an assembly or function. The levels progress from the more complex to the simpler divisions.

INDUCTIVE ANALYSIS: An analytical approach involving the evaluation of the parts of a system to determine characteristics of the whole system.

INTERFACE: The means by which equipment is interconnected.

INTERFACE FAILURE MODE AND EFFECTS ANALYSIS (INTERFACE FMEA): An analysis which determines the failure effects of failure modes within the interconnections between interfacing hardware or software elements in a system or subsystem. The interconnections may be signals, cables, wires, fiber optics, hydraulic lines, etc.

2.4 (Continued):

INTERFACE FAILURE MODE, EFFECTS & CRITICALITY ANALYSIS (INTERFACE FMECA): An extension of the Interface FMEA to include an assessment of the failure mode severity and probability of occurrence.

ITEM: A non-specific term used to denote any product, including systems, subsystems, parts, components, subassemblies, sets, accessories, software, etc.

LATENT FAILURE: A failure for which the annunciation to the operator is delayed beyond some acceptable threshold such that its occurrence can not be circumvented or corrected.

LINE REPLACEABLE UNIT (LRU): A piece of equipment that is removed and replaced in the field to get the system up and running again. Actual repair may take place elsewhere.

LOCAL EFFECT: The consequence a failure has on the operation, function, or status of the specific item being analyzed. See Failure Effect.

LOWEST LEVEL INDENTURE: The simplest division of the specific item being analyzed, usually a piece-part. See Indenture Level and piece-parts.

LRU: See Line Replaceable Unit.

MAINTENANCE LEVELS: The levels within a hierarchically organized maintenance organization at which different repair operations are performed. Typically, Line Level Maintenance refers to line maintenance operations such as the Removal and Replacement (R&R) of an LRU; Shop Level Maintenance refers to repair center (or depot) operations such as R&R of SRUs from an LRU; R&R of piece-parts from an SRU; or checkout of a refurbished SRU.

MISSION: The objective or task, together with the purpose, which clearly indicates the action to be taken.

MISSION IMPACT: The “Go/No Go” failure assessment concerning the ability of the system to operate or complete its mission. This is derived from the end-level effects the failure has on those major subsystems required by the system to operate or complete its mission. See End Level Effect, Failure Effect.

MODULE: A self-contained unit or subsystem that performs a specific task or class of tasks in support of the major function of the system.

NEXT HIGHER LEVEL EFFECT: The consequence a failure has on the operation, functions, or status of the items in the next higher indenture level above the indenture level under consideration. See Failure Effect.

OPERATING MODE: A state or phase of operation in which a system or subsystem can exist. The types of modes examined during the analyses can include normal operating modes, test modes, and transitions between modes, as defined in the applicable system requirements documents.

2.4 (Continued):

PART FAILURE NUMBER: A unique identifier for an individual failure mode.

PIECE-PART: The lowest indenture level hardware item at which a FMEA is usually performed.

PIECE-PART FAILURE MODE & EFFECTS ANALYSIS (PIECE-PART FMEA): An analysis that assesses the failure causes and effects of failure modes on the physical piece-parts which comprise the SRUs and LRUs of a system or subsystem.

PIECE-PART FAILURE MODE, EFFECTS & CRITICALITY ANALYSIS (PIECE-PART FMECA): An extension of the piece-part FMEA to include an assessment of the failure mode severity and its probability of occurrence.

RELIABILITY: The probability that an item will perform its intended function for a specified interval under stated operational and environmental conditions.

SCHEMATIC BLOCK DIAGRAM: A diagram showing the interrelationships of components or piece-parts of a system, subsystem, or function.

SEVERITY: The consequences of a failure mode as determined by the degree of injury, property damage, or system damage that could ultimately occur. Severity considers the worst potential consequence of a failure. See Severity Classifications.

SEVERITY CLASSIFICATIONS: Severity classifications are assigned to provide a qualitative measure of the worst potential consequences resulting from design error or item failure.

SHOP REPLACEABLE UNIT: A shop level maintenance item located within an LRU.

SINGLE POINT FAILURE: A failure of an item which is not compensated for by redundancy or alternative operational procedures that results in adverse end level effects.

SRU: See Shop Replaceable Unit.

SUBSYSTEM: A combination of components which perform an operational function within a system. Any indenture level or logical (by function) partitioning of the system being analyzed. A subsystem may be implemented physically as one LRU, or it may contain many LRUs.

SUPPLIED REQUIREMENT: A requirement placed on the end-item system by the procuring activity and supplied to the contractor in the product specification or statement of work.

SYSTEM: A composite of equipment, skills, and techniques capable of performing or supporting an operational role. The term system is commonly used to refer to the highest level of requirements and resource grouping applicable to the particular analysis being performed. The actual analysis for some programs could be at the system segment level (e.g., engine system).

2.4 (Continued):

SYSTEM INDICATIONS: The response of the system monitoring provisions that provide the operator with information about the system state, including performance, operational, and failure conditions.

UNDETECTED FAILURE: A postulated failure mode for which there is no system indication.

VALIDATION: The process of confirming on an authoritative basis to accepted engineering principles.

VERIFICATION: (a) The process of proving that a product complies with its formally established requirements. (b) The process of proving by special engineering inspections, analyses, demonstrations, or tests that a system, segment, or configuration item satisfies the requirements of the pertinent development specification.

3. INTRODUCTION:

Failure Modes and Effects Analysis (FMEA) is a formal and systematic approach to identifying potential system failure modes, their causes, and the effects of the failure mode occurrence on the system operation. FMEA provides a basis for identifying potential system failures and unacceptable failure effects that prevent achieving design requirements from postulated failure modes. A FMEA is used in many system design analyses including assessing system safety, planning system maintenance activities, defining provisions for fault recovery, fault tolerance, and failure detection and isolation, and identifying design modifications and corrective actions needed to mitigate the effects of a failure on the system. When the analysis is extended to compare failure modes according to the combined influence of the severity of their effects and the probability of occurrence of the effect, it is called a Failure Mode, Effects, and Criticality Analysis (FMECA).[1]

Early identification of design deficiencies minimizes the potential cost of necessary design changes. The objectives and uses of a FMEA include:

1. enhancing system safety by uncovering failure modes that result in hazardous conditions;
2. assessing the mission related effects of critical and/or undetectable failures;
3. influencing the design to mitigate the impact of failures on the final product;
4. supporting verification processes;
5. assuring that the Fault Detection/Fault Isolation (FD/FI) capability designed into the system will meet the FD/FI requirements imposed by the end-item specifications;
6. assisting the design engineer to select a design with a high probability of operational success; and
7. providing data for development of effective maintenance support.

[1] Throughout this document the term FMEA is used in a generic sense to include both the failure modes and effects analysis and the prioritization if that is done.

3. (Continued):

FMEA efforts and the design disciplines must work together to correct design deficiencies and mitigate the consequences of failures. When failures are found that present unacceptable consequences, the design is modified to comply with requirements, or derived requirements are fed back into the design to fix or improve it. By focusing attention on the weaknesses of a design and on what can go wrong in building and supporting a product, the FMEA plays a central role in improving the product design.

3.1 Overview of the Process:

FMEA is an inductive analysis that determines how every possible item failure mode affects the system operation. Failure modes are postulated for each item and the effects are traced through the system to determine the resulting system operation. The FMEA interfaces with the System Design, Detailed Design, and Reliability Prediction tasks. Each of these tasks provides inputs to the FMEA. The System Design task provides the requirements baseline and end-item architecture. The Detailed Design task implements the baseline requirements. The Reliability Prediction task provides an assessment of the failure frequency of occurrence that the design is expected to experience in the field due to the given failure mode.

As shown in Figure 1, the FMEA process is initiated during the program Conceptual Design phase and continues through the Preliminary Design phase to verify the adequacy of requirements. During the product's early development there is a great deal of uncertainty: customer needs are not fully understood, and requirements and specifications are usually incomplete. This inherent uncertainty is gradually resolved as the design develops from conception to completion. Refinements to the FMEA track the design as it matures. In the Preliminary Design phase Functional Analysis of the end-item facilitates the reduction of the design uncertainty by assessing how the item functions, both fault-free and with failures. In the Detailed Design phase the analysis is used to verify design compliance with requirements. Testing during the Verification and Validation phase provides a measure of the accuracy of the analysis and helps to maintain the integrity of design changes. Field data for assessing the accuracy of the analysis and maintenance troubleshooting procedures is collected during the Product Use and Support phase.

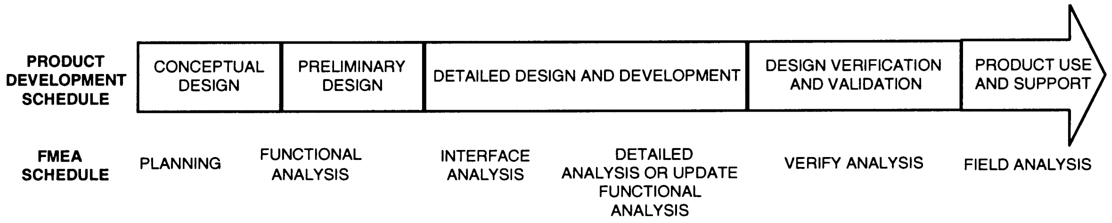


FIGURE 1 - Typical Product Development Cycle Versus FMEA Schedule

3.1 (Continued):

To achieve the objectives of the FMEA process, several analysis tasks are performed. These include:

1. Analysis Planning, initiated during the conceptual design phase;
2. Functional Requirements Analysis, initiated during the conceptual design phase;
3. Failure Mode and Effects Analysis which consists of:
    a. Functional Analysis, initiated during the conceptual or preliminary design phase;
    b. Interface Analysis, initiated during the preliminary or detailed design phase; and
    c. Detailed Analysis, typically initiated during the detailed design phase. In some cases the Functional Analysis may be updated during the detailed design phase rather than do a Detailed Analysis.
4. FMEA verification, initiated during the design verification and validation phase.

Figure 2 shows the general flow of the FMEA process. (The numbers in the boxes indicate the sections in which the indicated tasks are discussed.)

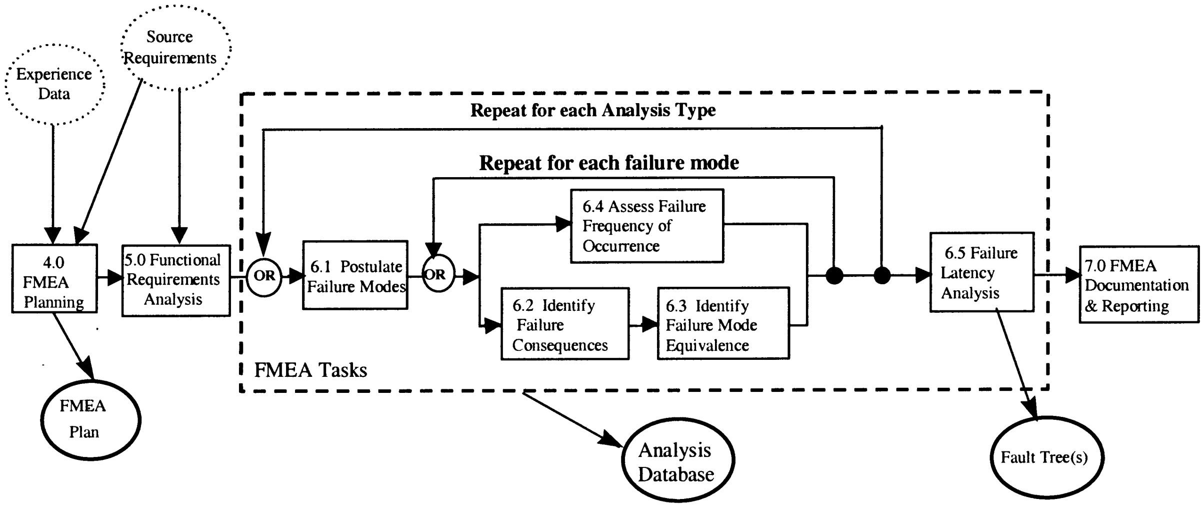

Source Requirements
Experience Data
Repeat for each Analysis Type
Repeat for each failure mode
6.4 Assess Failure Frequency of Occurrence
4.0 FMEA Planning
5.0 Functional Requirements Analysis
OR
6.1 Postulate Failure Modes
OR
6.2 Identify Failure Consequences
6.3 Identify Failure Mode Equivalence
6.5 Failure Latency Analysis
7.0 FMEA Documentation & Reporting
FMEA Tasks
FMEA Plan
Analysis Database
Fault Tree(s)

FIGURE 2 - Diagram of the FMEA Process

3.1 (Continued):

Before beginning the analysis the system requirements and operating concept must be specified to the degree that they are known. These include the system operating modes and functions, required performance levels, environmental considerations, and safety or regulatory requirements. Data such as field reports, design rules, checklists, and other guidelines based on lessons learned, technology advances, and the history or analyses of similar systems are collected and studied.

Analysis planning is performed to identify the use of the analysis results to provide design guidance, the procedures for verifying specified requirements and design compliance to requirements. The analysis process is identified to establish how the process and methodology complies with internal guidelines and specified standards, requirements and objectives. (See Section 4.)

Models (often in the form of block diagrams) are developed to illustrate the physical and functional relationships between items and interfaces within the system. From these models, come postulated failure modes to be analyzed for their effects on the system. Potential design weaknesses are identified by assessing the consequences of each failure, and the failure's effect on the system safety, readiness, mission success, demand for maintenance, or logistic support. Ways of detecting the failures and any compensating provisions or design changes needed to mitigate their effects are identified and prioritized based on each failure’s severity and probability of occurrence.

A FMEA is classified as either functional, interface, or detailed[2] according to the way in which the failure modes are postulated. Each type represents a refinement of the system model. As the design details become available the analysis iterates and the system models are expanded and refined until the system has been completely defined, analyzed and documented. The functional, interface, and detailed analyses, combined with the FMEA planning and Functional Requirements Analysis form the basis for the FMEA process outlined previously. These different analysis types are performed during the various phases of the product development cycle. Conducting the analysis in this iterative manner at increasing levels of detail enforces a disciplined review of the baseline design and allows timely feedback to the design process.

3.1.1 Functional FMEA: A Functional FMEA is performed on the conceptual design to support the architectural definition and verify necessary design compensation and failure recovery requirements derived by the Functional Requirements Analysis. A Functional FMEA is “black box” in the sense that the functions an item performs are analyzed rather than the characteristics of the specific components used in its implementation. Functional FMEAs may be performed on control systems, processes, software, and complex devices, whose functionality are more readily understood than the details of their operation. A Functional FMEA benefits the design by influencing the definition of the design prior to constructing hardware, coding software, or implementing a process.

[2] When applied to hardware, the "detailed" FMEA has historically been called a "piece-part" FMEA. The new terminology recognizes that the FMEA can be applied to a broader range of applications.

3.1.1 (Continued):

A Functional FMEA is performed following guidelines developed in the analysis ground rules. A functional analysis focuses on the functions that an item, group of items, or process performs rather than the characteristics of the specific implementation. The analysis begins with a functional block diagram or equivalent system representation.

The functional block diagrams of the system elements are examined to identify specific types of functions. Each function type identifies a set of previously postulated functional failure modes. Each function is analytically failed, one at a time, in its respective failure modes. The fault characteristics (effects and fault signatures) of each failure mode on the subsystem(s) are determined for each applicable operating mode. As more design details become available, the functional block diagrams and analyses are refined. The analysis iterates in this manner until all the system elements are completely defined and documented. Any single-point failures or undetected failures that cause undesirable end-effects such as loss of end-item functions are reviewed with the respective design discipline. Deficiencies are corrected by making design modifications to the conceptual design or the baseline requirements. The Functional FMEA is then revised to reflect the modifications.

The Functional FMEA is documented through a collection of analysis worksheets. Characteristics of the postulated failure modes are captured on worksheets, usually on a subsystem basis. The worksheets organize all failure modes that exhibit identical consequences into fault equivalence groups.

3.1.2 Interface FMEA: An Interface FMEA is performed in the same manner as the Functional FMEA to verify compliance to requirements. Interconnections between subsystems, particularly those designed by separate design groups provide the basis for the postulated failure modes. The advantage of a separate Interface FMEA is that it can be performed before the detailed design of the interconnected subsystems is available. It is begun as soon as the system interconnections are defined to assure that the proper interface protocols are designed. Typical outputs of this analysis are interface failure modes that need to be eliminated or mitigated by interface design changes.

The Interface FMEA is the process of determining and recording characteristics of failures in the interconnections between interfacing system elements. The Interface FMEA begins with the gathering of information on the interfaces of system elements. As shown in Figure 3 interconnections between interfacing hardware elements may be cables, wires, fiber optic lines, hydraulic lines, pneumatic lines, etc. Additionally, information on the software interfaces between system elements should be identified. Previously obtained or generated unit interface diagrams, interface requirements specifications, and interface control documents are examined to identify specific types of interfaces. Failure modes specific to the interfaces are defined and their characteristics (effects and fault signatures) are determined.

Each type of interconnect has its appropriate corresponding failure modes. For example, if the interface is an electrical power system cable, the effect of the “open” failure mode is considered for each wire composing the cable.

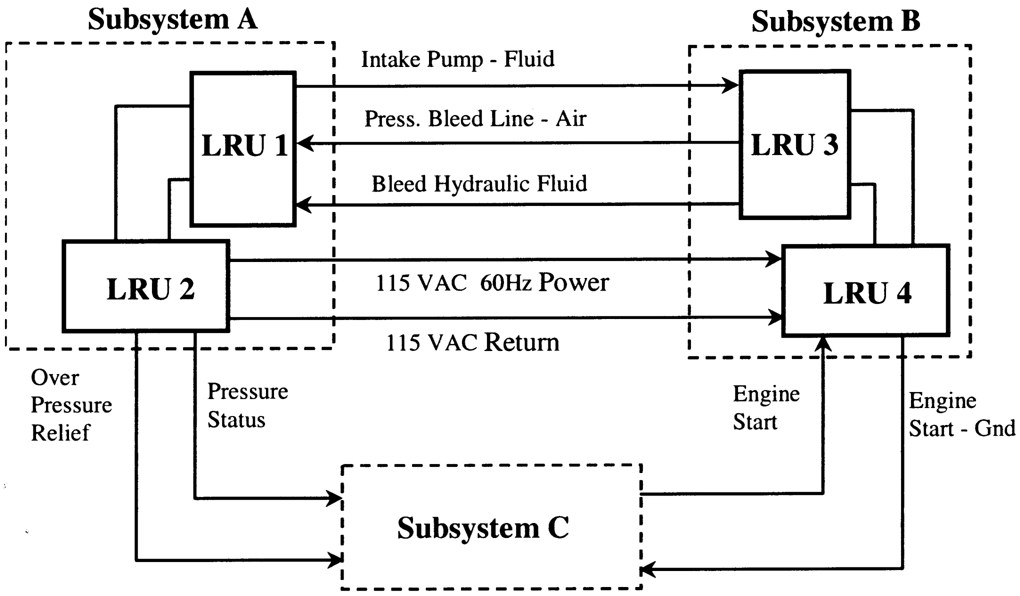


FIGURE 3 - Interface FMEA Examines the Interfaces Between Subsystems

3.1.2 (Continued):

For large systems the system integrator must ensure that the analysis is complete. The subsystem owner or designer is responsible for assessing the effects of all inputs to the subsystem. The system integrator uses the results of the subsystem owners' assessments and applies the results to the interface architecture accounting for all fan-in and fan-out conditions.

3.1.3 Detailed FMEA: A Detailed FMEA is performed in the same manner as the corresponding Functional FMEA to verify that the design complies with requirements for failures that can cause loss of end-item functions, single point failures, fault detection, and fault isolation. Components representing the implemented design provide the basis for the postulated failure modes. In a hardware Detailed FMEA the components comprise the physical system design. In a software Detailed FMEA the components are from the software source code.

A Detailed FMEA is the process of determining and documenting characteristics of the failures of each component within the individual system elements (piece-parts, functional part groupings, and as-implemented software). A Detailed FMEA is initiated as the design of each element matures and the detailed design schematics, parts lists, and detailed software design documents and source code become available. This analysis can be performed on the engineering model and revised for changes reflected in the system operational model. An established set of failure modes is essential for performing the detailed analysis correctly. Any single-point or undetected failures that have undesired consequences are reviewed with the respective design discipline. Deficiencies found are corrected by the appropriate design group and the analysis is revised accordingly.

3.1.4 FMEA Verification: Verification of the FMEA is conducted as required during design verification and testing to verify:

1. end-item consequences match the analysis results;
2. circumventing actions identified in the analysis do, in fact, mitigate the failure effects; and
3. monitoring provisions correctly isolate the failed LRU as a possible cause of the system failure.

Generally, this verification uses the results of other tests rather than a separate test performed exclusively for the FMEA verification. Lessons learned are captured for inclusion in the FMEA planning for future programs. Test results form the basis for changes to ground rules and subsequent iterations of the analysis.

The FMEA continues in effect throughout the life of the program. Applicable test results and field data are periodically collected and reviewed for deviations from the analysis results. The analysis is updated as required to reflect the failure consequences captured in available fault insertion tests and operational field data which conflict with the analysis results. Design changes are assessed to determine if an analysis update is necessary. Additional analysis is required to:

1. assess new failure modes and their consequences;
2. verify compensating provisions are maintained;
3. revise the fault isolation procedures as applicable; and
4. maintain the analysis database to reflect the baseline configuration.

3.1.5 Documentation: When the design is complete the FMEA documentation provides a record of the analysis. The documentation includes a description of the system or end-item being analyzed, the collection of analysis worksheets resulting from the functional, interface, and detailed analyses, and a summary of the analysis results. Undetected, but manageable failure modes within the system are identified by the analysis, categorized, and their manageability is explained.

3.2 FMEA Applications:

FMEAs have been applied to a variety of products from a single item to complex systems containing thousands of parts. Recent applications of FMEA have extended the analysis to the processes by which a product is built (Reference 2.1.4) or a service is provided, and to the software that provides product functionality (Reference 2.3.12). A process FMEA is analogous to a product FMEA in that the process is subject to the design development cycle of “define”, “design”, and “verify”. A product FMEA analyzes the product design by examining the ways that item failure modes (hardware and software) affect the product operation. A process FMEA analyzes the processes involved in manufacturing, and assembling the product by examining the ways that failures in those processes affect the operation and quality of the product.[3] Figure 4 illustrates the types of FMEA analyses that can be applied to products (hardware and software) and processes.

[3] These have previously been called "design FMEA" and "process FMEA" respectively. The new terminology recognizes that both types of FMEAs focus on design – design of the product or design of the process – and emphasizes whether the focus of the analysis is on the product itself or a process associated with the product.

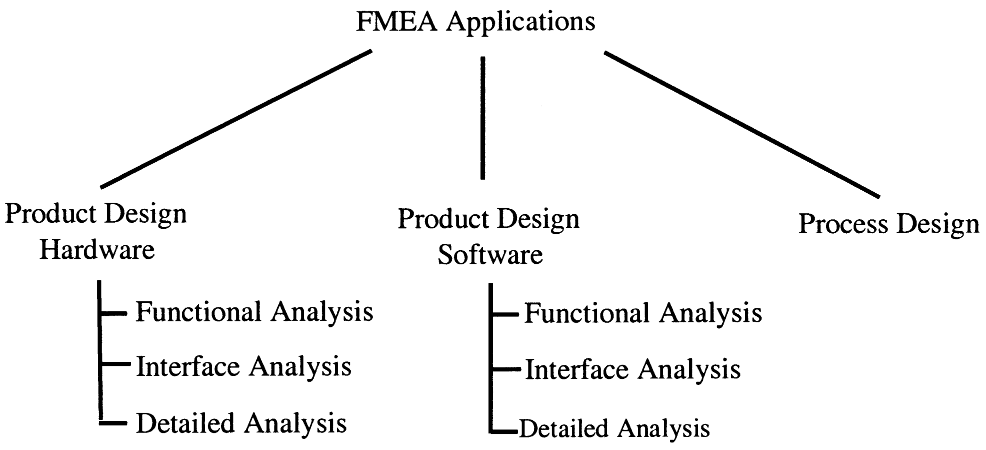


FIGURE 4 - Applications of FMEA

3.2.1 Product Design Hardware FMEA: Product hardware FMEAs are applied to the physical design of the product. Hardware includes, for example, a system's electrical, mechanical, and hydraulic subsystems, and the interfaces between those subsystems. The analysis can be done in terms of failures of the functions performed by the hardware, the interfaces between hardware modules, or the individual components from which the hardware item is implemented.

3.2.1.1 Hardware Functional FMEA: A hardware Functional FMEA is first applied early in the design cycle, after the major system functional components and their interactions have been defined – most often as a functional block diagram. Failure modes appropriate to each functional component are defined and their failure effects determined by logically simulating the system at the functional level. The usual functional failure modes are that a particular function is not performed or it is performed incorrectly. Assessment of the effects of these failure modes allows the analyst to identify weaknesses in the design or establish the need for redundant subsystems to eliminate single points of failure. Design corrections are effected by changes in the system requirements.

When performing a Functional FMEA the analyst must be aware of the environment in which the equipment performing the function operates since the system operating mode and the states of functions at higher indenture levels may alter the behavioral characteristics of the function under analysis. For example, a failure within an airplane's landing gear will not have an adverse impact to the plane during the "cruise" mode, but it may have significant adverse effects during the "landing" mode. Failure consequences that are different for different modes of operation must be captured for all relevant states.

The hardware Functional FMEA is also used later in the design cycle for complex subsystems or components such as integrated circuits and control systems whose functionality is more readily described than the operation of individual components.

3.2.1.2 Hardware Interface FMEA: A hardware Interface FMEA is performed on the physical interfaces between major functional system elements. (Such elements are typically referred to as LRUs (Line Replaceable Units) in a hierarchically organized maintenance organization.) These units are often designed within a single design group and the input and output interfaces with other system elements are usually specified much earlier in the design cycle than the internal design of the unit. Depending on the type of system, the unit interfaces typically involve mechanical linkages, hydraulic lines, or electrical cabling. The failure modes considered are those appropriate for the type of interface and the level to which the design has progressed. For example, grounding of an electrical cable or low pressure in a hydraulic line are appropriate failure modes early in the design cycle, a bent pin shorting a signal conductor in an electrical cable is more appropriate late in the cycle. The results of the FMEA document the types of hardware malfunctions anticipated at the unit interface and generate design requirements for the unit to be able to withstand, correct, or compensate for the failure of the interface without doing further damage.

3.2.1.3 Hardware Detailed FMEA: A hardware Detailed FMEA has traditionally been the most common type of FMEA application. It is done at the lowest piece-part level of design and generally involves individual system components. Standard lists of potential failure modes are available for many of the most used components (Reference 2.3.4). Since individual components are involved, the design must be nearly complete before the analysis can be done. Furthermore, in large systems circuit simulation software must often be used to determine the system level effects of the failure.

Where the expected failure effect is transient, in response to temporary hardware disturbances (e.g., EMI, Power noise) or to software failures, the symptoms that are available to any automatic detection scheme and the data that any automatic fault detection equipment might record for later fault isolation should be identified. If the transient effects of these failures are identifiable by the system operator(s), the expected system behavior, including recovery behavior, should be noted for later inclusion in operations manuals.

3.2.2 Product Design Software FMEA: Software includes programs, their related data elements, and their execution as tasks that implement various system functions. Software also includes program interfaces with the hardware (HW Interfaces) and the interfaces between different programs or tasks (Software Interfaces). Software can be embedded as a functional component in a self-contained system or executed on a general purpose computer.

Software FMEAs, as with all FMEAs, tend to be labor intensive. Consequently, it is desirable in most cases to limit software FMEAs, particularly software Detailed FMEAs, to those elements of a software system whose correct functioning is crucial for continued safe (or revenue critical or other significant criteria) operation. It is crucial in developing limits on the scope of the software FMEA that the analyst thoroughly understands the entire software design. Identifying which software elements need to be analyzed requires that the analyst understand both the critical software elements and the protection (e.g., firewalls) provided to ensure that the critical software elements cannot be corrupted by data provided from non-critical software elements. Software FMEA costs can be minimized if the software system has been designed with effective partitioning between critical and non-critical software elements.

3.2.2 (Continued):

In addition to thoroughly understanding the software being analyzed, the analyst performing a software FMEA needs to have detailed knowledge of the underlying hardware, the software language being used, and the specifics of the software development tools being used. Analysis of software, an abstract representation of a logical design (functional level) and of the specific machine instructions (detailed level) to be implemented, has a major assumption embedded in the technique. Analysis of the software design assumes that the design as represented in design documents, pseudo-code, and later high-level language code, is an accurate representation of the design which will actually be implemented in the system. This assumption, in turn, leads to requirements on the software design environment that must be instituted to allow software FMEA results to be valid. The development program must be limited to the use of compilers that are certified to the standard for the language being used. This implicitly limits the set of possible development languages to those which have formal standards and for which certified compilers are available. Additionally, optimization features within the compiler must either be disabled, or their impact must be completely understood. The language used within the development program must also be limited to elements of the language that are defined within the language standard. Those language features that are left to the compiler developer to completely define need to be avoided in any development, which will use FMEA as part of its verification process. Thus, involvement of the organization tasked to perform software FMEA early in the software development process is crucial. Early design involvement will allow needed compiler and language restrictions to be imposed on the design process at a cost-effective time.

3.2.2.1 Software Functional FMEA: A software Functional FMEA is applied to the Computer Software Configuration Item (CSCI) during top-level software design. Failure modes appropriate to the individual functions which have been assigned to Computer Software Components (CSCs) and to individual modules are developed and applied to the software design to determine the effect on system performance and safety of incorrect performance of the software element. Assessment of the impact of potential software failures allows the analyst to identify software architecture vulnerability to single point failures and timing dependencies that could lead to unintended effects. The primary outputs of the Software Functional FMEA are used to identify software (and possibly system hardware) architectural changes to reduce failure exposure and to identify requirements to ensure that incorrect software behavior can be detected and appropriate system corrective actions instituted. Software Functional FMEA at the system level is crucial to determining the ability of fault tolerant designs to gracefully handle failures under conditions of possible data corruption.

3.2.2.2 Software Interface FMEA: A software Interface FMEA is similar to a Functional FMEA for software, but focuses on the interfaces between disparate software and hardware elements. It is often limited to large systems – those that communicate values and/or commands between distinct system elements with separated but cooperative software functionality. Failure modes specific to the message and/or data type being passed are postulated and the system level effects identified. Causes of incorrect message contents can include software failures, transient external faults whose effects are coupled through hardware, and transient and permanent hardware faults that could corrupt message contents. The results of the analysis are used to help identify software vulnerability to external message contents (including sensor errors where appropriate) and to allow development of software requirements and robust software designs which minimize or eliminate unacceptable vulnerabilities to potentially corrupt external data. Assessment of this vulnerability is particularly important when applied to fault tolerant designs.

3.2.2.3 Software Detailed FMEA: Software Detailed FMEAs are designed to evaluate the impact of single variable or instruction failures as such failures propagate through the software. As such, software Detailed FMEAs should generally be reserved for systems that do not include robust hardware protection of memory elements, processing results, and data transfers. Systems in which the processing hardware is designed to ensure that all plausible memory failures, data transfer failures, and erroneous processes or results are detected prior to variable or instruction use and either corrected (for example by Error Detecting and Correcting Coded (EDCC) memory) or reported, leading to “safeing” of the system, will usually not benefit sufficiently from the results obtained by performing a software Detailed FMEA to justify the associated costs. Systems in which the processing hardware does not have effective memory protection, processing results protection (e.g., arithmetic residue codes), and memory transfer protection (e.g., parity or Cyclic Redundancy Code (CRC) checks on bus transfers, including transfers to storage elements) should be considered candidates for a software Detailed FMEA. A software Detailed FMEA can provide considerable insight into the behavior of embedded control systems with minimal hardware protection including those which depend on periodic tests to validate the correctness of the system hardware and software.

A software Detailed FMEA is performed as a part of detailed software design. The analysis is based on the software top level design documents, detailed design documents, and source code listings. The intent of the Detailed FMEA is to supplement the Functional and Interface FMEAs with a detailed assessment of the response of the as-developed software to plausible faults and failures. Both the Functional and Interface FMEAs will often need to be updated at the time the software Detailed FMEA is performed to reflect the ‘final’ software architecture. The Detailed FMEA is based on applying a logically complete set of potential failure modes to each software module’s functionality and to the variables used by the module. The failure of the detailed module functionality is examined to determine the system effects of the fault or failure. Similarly, the failure of a variable to correctly maintain value is also examined for its system effects.

3.2.3 Process Design FMEA: A process FMEA assists in the analysis of manufacturing and assembly processes. The process FMEA assumes the product, as designed, will meet the design intent provided the product is manufactured properly according to its specifications. FMEAs are applied to processes to focus on potential product failure modes that result from manufacturing or assembly process deficiencies. Identification of these process deficiencies enables engineers to:

1. identify potential unacceptable customer effects of manufacturing failures;

2. develop monitors or control procedures to reduce the frequency of producing unacceptable products or increase the detection of unacceptable products;

3. develop thorough Manufacturing Control Plans;

4. establish a priority for process improvements; and

5. document the rationale behind process changes to guide development of future manufacturing and assembly processes.

Process FMEAs are conducted for new parts and processes, changed parts and processes and new applications and environments for product manufacturing and assembly. Process FMEAs are initiated prior to tooling for production and take into account all manufacturing operations from individual components to assemblies.

Outputs from a process FMEA include:

1. a list of potential process failure modes;
2. a list of recommended corrective actions;
3. a list of process modifications to eliminate the causes of failures, or reduce the frequency of their occurrence, and improve the defect detection capability of the manufacturing system.

Early analysis review is conducted to anticipate, resolve and monitor potential process concerns during the manufacturing planning stages of a new part or product.

3.3 Cautions:

Several cautions should be observed in the application of a FMEA. First, a FMEA traditionally considers only non-simultaneous failure modes. Each failure mode is considered individually, assuming that all other system components are performing as designed. Hence, a typical FMEA provides limited insight into the following anomalous behaviors:

1. the effects of multiple component failures on system functions, and
2. latent manifestations of defects such as timing, sequencing, etc.

Other analysis techniques, such as Fault Tree Analysis, Sneak Circuit Analysis, Markov Analysis and computer-aided simulations, are helpful in overcoming these limitations.

3.3 (Continued):

Second, the prioritization of the failure modes for corrective actions is substantially subjective. Thus, care should be taken in decision making when using any quantitative aspects of the numbers presented in the analysis.

4. FMEA PLANNING:

The benefits of a FMEA depend upon whether the analysis can be completed within the proper time frame and at a reasonable cost. If the analysis results are too late then design deficiencies become too costly to correct resulting in reduced product quality or performance. The analysis itself then becomes costly, as it does not add value to the product. Therefore, FMEA planning should include estimating:

1. the costs required for implementing the FMEA process;

2. the impact on the product development costs of such expenses as late design modifications or over design; and

3. the impact on the operational costs of the product due to factors such as the product maintainability and its reliability.

Planning the analysis work includes defining procedures for verifying specified requirements, using the analysis results to provide design guidance, verifying design compliance to requirements, and updating the analysis to reflect design changes. The product requirements, objectives, complexity and criticality of the system functions are assessed for each subsystem to determine the analysis tasks, depth of analysis and final output products. Using a team approach, with members from design, manufacturing, reliability, test, logistics, and quality assurance groups, may result in a more complete and thorough analysis. The team approach ensures integration of the product and process planning and provides for communication between organizations.

Commensurate with program business practices the FMEA planning will result in a documented plan. The plan identifies the FMEA inputs, outputs, program support and any tailoring required to support the program requirements. Identification of the initial end-item attributes serves as a guide for developing the scope of the FMEA activity including the analysis boundary conditions, end item functions, interfaces, ground rules and products. An overview of the analysis database is provided by the system integrator to identify which fault characteristics will be assessed. The overview identifies what data items will be tracked and what output reports are required to support the analysis results. The program schedule for completing the tasks is identified along with an estimate for any technical coordination meetings to review the analysis.

4. (Continued):

Other items typically included within the FMEA plan are:

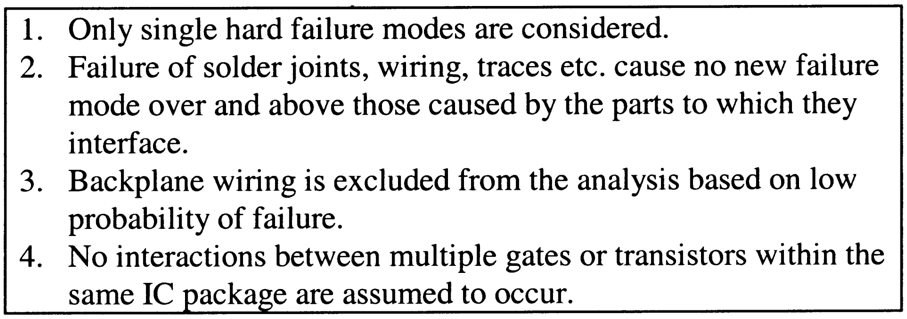
1. the ground rules, analysis assumptions, trade study results, coding system description, failure definitions, glossary of terminology, and worksheet formats;
2. the end effects to be assessed for each fault condition;
3. severity classification definitions related to the documented end effect; and
4. the data interchange procedures (specifying content, format, and media) to be used between interfacing processes and customers.

4.1 FMEA Ground Rules and Assumptions:

The complexity of the failure analysis tasks makes it essential to adhere to a well-defined and structured process. Therefore an initial set of ground rules and assumptions must be established prior to commencing the FMEA. Changes to the analysis ground rules and/or assumptions necessitate a reassessment of the analysis results to identify needed updates.

General ground rules, not influenced by the characteristics of any particular end-item, identify: assumptions, limitations, analysis approach, boundary conditions, failure criteria for fault models (functions, unit interconnects, and components), and what constitutes a failure (in terms of performance criteria, success/failure criteria, interface factors, or coding schemes). Table 1 lists examples of some general ground rules.

TABLE 1 - Example Ground Rules



1. Only single hard failure modes are considered.
2. Failure of solder joints, wiring, traces etc. cause no new failure mode over and above those caused by the parts to which they interface.
3. Backplane wiring is excluded from the analysis based on low probability of failure.
4. No interactions between multiple gates or transistors within the same IC package are assumed to occur.

4.1 (Continued):

Ground rules specific to the end-item application identify: applicable source requirements to be verified, end-item equipment to be analyzed (e.g., operational, ground support, maintenance support, special test equipment, etc.), lowest indenture level at which the analysis will be done, environmental conditions assumed, primary and secondary mission objectives, modes of operation, criteria for analyzing an operating mode, and accident risk factors as defined by system safety analyses. Identification of the product characteristics serves as a guide for developing the scope of the FMEA. At the beginning of the conceptual design phase, analysis requirements and objectives are established by evaluating: source requirements, operational objectives, statements of work, maintenance concepts, and contractual requirements.

4.2 Analysis Tailoring:

The diversity of end-items, in terms of requirements, complexity and criticality, drives a need for a tailorable FMEA process. Objectives such as minimizing analysis cost and achieving design requirements necessitate tailoring the analysis process to the needs of the program. For example, in considering a weapons system, a surveillance system, and a training system, a more comprehensive effort is normally required for the weapons system and a less comprehensive effort for the training system. Analysis tailoring is conducted to provide a FMEA process that identifies design deficiencies in a timely manner so that corrective actions can be implemented or compensating provisions established. The tailoring addresses both the end-item modeling and the analysis procedures to be followed.

Analysis modeling includes:

1. establishing the lowest indenture level of the FMEA for each program design phase; and
2. identifying the end-item boundaries.

Analysis procedures include:

1. allowances for analysis tasks, to be added or deleted, (e.g., functional requirements analysis, identification of failure causes, etc.);

2. coordinating the analysis content with the equipment/system supplier or customer/integrator (e.g., inclusion of end-level effects by the supplier);

3. establishing the criteria for performing each analysis type (i.e., functional, interface, and detailed); and

4. allowing for empirical analyses when qualitative or quantitative results are not essential or not timely for a particular design.

4.2 (Continued):

To achieve these objectives necessitates an organized methodology for applying analysis techniques and strategies to satisfy the technical objectives of the analysis. The result of the analysis tailoring is a set of boundary conditions captured in the analysis ground rules that define the analysis scope. Therefore, it is essential that the FMEA planning include how the analysis process is to be tailored for the end-item under analysis.

4.2.1 Depth of Analysis: The level of analysis (or the indenture level in a hierarchical decomposition of the design) at which a FMEA is performed applies to the level at which failures are postulated. A FMEA can be performed at any level, from the overall system level to the lowest component level. Generally, the lower the level of indenture at which the analysis is performed, the higher the fidelity of the analysis.

Table 2 provides a summary of the significance of the item failure and the corresponding level of detail required of the analysis. For mission essential items, a Detailed FMEA only needs to be provided for those functions whose functional analysis indicates a need for high safety or high reliability.

4.3 Supplier/Subcontractor Integration:

Integration of the results of a supplier or subcontractor FMEA with a system level FMEA is best supported with a “seamless” approach that allows the FMEA at lower levels to be “rolled up” to higher indenture levels without increased ambiguity. The criteria for a “seamless” analysis includes:

1. common understanding of the analysis content (Note: this should be agreed to by all parties as part of the FMEA plan);

2. common nomenclature for failure modes and consequences;

3. consistent application of failure mode failure rates and probabilities;

4. a glossary of analysis terms (usually provided by the integrator); and

5. agreed to definitions of how the analysis data will be provided to the integrator (including any electronic data interchange requirements).

Ground rules, nomenclature libraries, and data interchange procedures are provided to all suppliers or subcontractors to support the roll up of their results.

TABLE 2 - Tailoring Guidelines

FMEA Task	Value/Use	Timing	Recommendations
Functional Requirements Analysis	Defines the design requirements for fault compensation, mitigation and monitoring provisions.	Initiated during conceptual design phase.	Should always be performed.
Functional Failure Mode and Effects Analysis	Supports functional assessment of system architecture. Supports early verification of the conceptual baseline: • Completeness of fault compensation requirements; • Requirements for FD/FI provisions. Identifies critical functions for more detailed analysis.	Initiated during conceptual or preliminary design phase.	Should always be performed.
Interface Failure Mode and Effects Analysis	Supports system level assessment of down-stream failure effects (e.g., cascading faults). Provides a system view to the response of the FD/FI provisions. Provides an assessment of the overall system architecture.	Initiated during preliminary or detailed design phase.	Performed when analyzing a system or subsystem or when required by the system integrator.
Hardware Detailed Failure Mode and Effects Analysis	Provides a higher fidelity assessment for critical and safety related functions. Provides a detailed assessment of LRU and SRU failure conditions.	Initiated during detailed design phase.	Should be limited to safety or mission critical functions identified during the Functional Failure Mode Analysis.
Software Detailed Failure Mode and Effects Analysis	Provides evaluation of single variable or instruction failures in software.	Initiated in detailed software design phase.	Should be limited to systems without hardware protection of memory, processing results, or data transfers.
Latency Assessment	Accounts for multiple simultaneous failure modes.	Performed as part of each analysis type.	Performed when there are safety concerns.
FMEA Verification	Verifies accuracy of analysis results. Validates analysis ground rules.	Initiated in verification and validation phase.	Done in conjunction with system verification testing, especially when the analyst is uncertain of the failure consequences, or when required by contract or concerned about ground rules.

4.4 Analysis Maintenance:

Analysis evaluations of product changes are required throughout the product life cycle to ensure the integrity of the product safety, maintainability, and reliability. The FMEA methodology must minimize the sensitivity of the analysis to change. To achieve these goals the analysis procedures must address:

1. making the analysis results repeatable;
2. making the analysis traceable to program drawings, documentation, and other analyses;
3. minimizing the impact of changes by partitioning the analysis; and
4. making global changes to the analysis from a single, central control point.

4.4.1 FMEA Repeatability: The FMEA methodology must support the ability to reproduce an analysis with the same results. This feature is required to withstand any challenges to the accuracy and integrity of the analysis. A repeatable analysis requires:

1. clear definitions of failure modes and consequences;
2. raceability of analysis items to program drawings, documentation other analyses; and
3. features (such as comment sections) that allow the analyst to capture discussions with program designers regarding subtle features and capabilities of the design.

4.4.2 FMEA Traceability: The final analysis must reflect the actual product as installed in the field. That is, the analysis results must be traceable to the configured product. Trace provisions within the FMEA are essential for eliminating re-work and reducing the analysis cycle-time. Some examples of traceability objectives are:

1. To support regulatory agencies (e.g., FAA/JAA, FDA, DOD/MOD, DOE), traceability that shows the design's compliance to requirements is needed. These requirements typically include the probability of unwanted end-effects, and the probabilities of successful application of fault detection / fault isolation (FD/FI) procedures. The analysis results typically show trace paths from the postulated failure modes to the system schematics, design drawings, and wiring diagrams. The analysis also identifies the source documentation for item failure rates and failure mode probabilities.

2. To support maintenance activities, traceability is needed to link the consequences of a failure mode to the perception of a fault by a mechanic. Typically, it is sufficient for the analysis to show traceability to the set of isolation procedures referenced by the mechanic.

3. Internally, traceability is needed to manage the constituent elements of the analysis data set. A traceable analysis is necessary to support an orderly approach to the design assessment, continued maintenance of the analysis from field results and design changes, and maintain a historical recording of all changes to the analysis (i.e., what changed and why it changed).

4.4.3 FMEA Coding: A coding system is necessary for consistently identifying system functions and equipment and for tracking failure modes and effects in the FMEA. Such a system is often based upon the hardware breakdown structure or a similar numbering system. A uniform identification code helps to provide visibility of each failure mode and its relationship to the system function identified in the applicable diagrams.

In Functional FMEA, numbering conventions such as 1.23.245 which identify the system (1), subsystem (23), etc. are often used in conjunction with a function number to uniquely identify each end-item function. In Interface FMEA, to-from conventions are often used in conjunction with an interface number to uniquely identify each interface. For example, an electrical interface may be identified as: From-LRU, From-Connector, From-Pin, To-LRU, To-Connector, To-Pin, In Detailed FMEA, numbering conventions containing a unit designator, component reference designator, and pin number are often used. (For example, U31-2, s-a-0 might identify pin 2 of device U31, stuck-at-0.)

4.5 Analysis Libraries:

For complex systems, the large quantity of FMEA data makes a systematic database approach essential for performing the analysis. A database structure using a reference table approach enforces a standardized methodology and consistency in terminology. These reference tables, called nomenclature libraries, allow storing data in one place to ensure completeness and facilitate the process of making global changes as the analysis evolves.

Nomenclature libraries permit the system analysts to store, modify and retrieve common FMEA data elements. The availability of data, the indenture level of the analysis, and the analysis approach influence the extent of the library’s development and its application. Nomenclature libraries provide direction as to the level of detail to be analyzed while promoting the use of uniform terminology and documentation. Nomenclature libraries developed for one end-item can often be carried over for the analysis of other end-items of the same general type. Initial versions of the following libraries are defined and may be included as part of the FMEA plan:

1. a common set of functional, interface, or detailed item failure modes;

2. a common set of mission phases and/or operating modes;

3. a common set of end-effects that describe the effects each failure mode has on the end-item;

4. a common set of next-level effects that describe the effects each failure mode has on the next upper indenture level above the postulated failure mode;

5. a common set of severity descriptions to classify each failure mode effect according to the significance on the end-item being analyzed; and

6. a common set of monitor descriptions that identify the means by which the presence of a failure mode is detected.

4.5 (Continued):

The libraries are updated throughout the analysis as new data item values are needed and become available.

The Failure Modes/Causes Library typically is system independent for low-level or off-the-shelf components, but the library is often customized for specific types of systems; higher level items are usually system dependent.

The following sources can facilitate developing FMEA libraries:

1. History Files: History files for similar systems are good sources for developing a library. Documented test data, failure information, and field data can be valuable sources of information. History files may be used to develop failure modes and causes. Field and test data, if available, can aid in the development or validation of failure detection methods since the way in which each inserted failure mode was detected is identified in the report.

2. Source References: Industry accepted documents containing generic failure mode distributions (e.g., FMD-97 (Reference 2.3.4), NPRD-95 (Reference 2.3.3), MIL-HDBK-338 (Reference 2.2.1)) are appropriate sources of item failure modes and causes. Data contained in the source references can be normalized to account for the most probable failure modes or for only hard failure modes, as appropriate, for the system being analyzed.

3. Completed FMEAs: FMEAs for parts of a system or similar systems, maintained in a library contain information useful to the remaining FMEA effort.

Engineering interpretation of the data is critical when past analyses are used as sources of failure modes and effects information. It is important that the analysis not become so automatic that new and different failure modes or causes are overlooked.

To avoid analysis errors, FMEA libraries should be reviewed for accuracy, applicability, and traceability. Care must be exercised when library data is used in multiple applications. Generic data in source references is often a composite representing many different device variations and many different applications. For example, a dc motor operating as a starter may have different failure modes and different failure mode distributions than a dc motor operating as a winch.

Initial versions of the libraries will not address all situations. Thus revisions to the libraries will be necessary throughout the FMEA process.

5. FUNCTIONAL REQUIREMENTS ANALYSIS:

Functional Requirements Analysis is an essential step in establishing a baseline design against which a FMEA can be done. Functional Requirements Analysis is performed by Systems Engineering to develop the system architecture and allocate performance requirements to the system components. Fault compensation and monitoring requirements, that avoid or mitigate critical failures of the end-item, are developed as part of this analysis.

5.1 Requirements Analysis:

Requirements are derived by decomposing the source requirements and refining the system-level requirements into a complete set of detailed technical requirements. The detailed requirements are then allocated to the system hardware and software elements to establish lower-level requirement baselines.

Development of the FMEA begins with an understanding of the design intent – a knowledge of what the design is expected to do as well as what it is explicitly prohibited from doing. Customer needs from sources such as Quality Function Deployment (QFD), government regulations and known product requirements are compiled to permit decomposition and allocation of the requirements among the system subfunctions. Conflicts between multiple requirements must be reconciled in accordance with the mission objectives and priorities. The resulting requirements form the starting point for the subsequent requirements analysis.

As a prerequisite to decomposing functions into their constituent elements, the interfaces between the functions must be identified and quantified. Specifically, the analyst must associate the inputs with the function and its outputs. This permits the requirements analysis to be performed at the boundary of the subsystem, by specifying the functions and their performance requirements for specific conditions at the subsystem interface, without regard to its internal design.

Requirements associated with mission safety, system reliability, system anomalies, and system maintenance are collected for further decomposition and allocation to the system or subsystem elements. These include requirements for mitigating faults, fault monitors, fault signal data processing, signal path interconnectivity, operational test, and fault display. These requirements also form the set of those to be verified by the FMEA.

Requirements are analyzed in light of the operations and maintenance concepts, including conditions of failure. Each individual requirement is stated in terms of “What”, “How Well” (i.e., performance), and “When” (i.e., specified conditions). Derived requirements arise from constraints, consideration of issues implied, but not explicitly stated by the customer or user, and factors introduced by the developer's unique business considerations, and regulations.

A functional model of each subsystem is prepared while developing the baseline requirements. A common format used to portray the functional model and perform the analysis is a functional flow diagram. This diagram depicts the process of requirements derivation from the functional performance definition.

Because the failure mode analyst is uniquely concerned with the system response in the presence of failure, the analyst may be the first to identify specific failure conditions that must be analyzed.

5.1 (Continued):

The treatment of anomalies ("what-if" conditions) must be included in the formulated requirements associated with every allocated function. The requirements are reviewed to verify that, for every function defined, there exists a companion function that treats abnormal or anomalous conditions. The system-level requirement to "compensate for anomalies" is levied as complements to individual functions. The "normal" condition is defined first; then the "anomalous" condition is treated. Failure compensation may include active or passive redundancy, monitoring provisions, alternative operating modes, operator procedures, or other ways to mitigate the impact of failure.

5.2 Requirements Allocation:

Requirements are allocated iteratively to flow down performance requirements and design constraints to lower level functional modules. In the process, successively lower-level functions required to satisfy higher level anomaly related requirements are defined.

While the systems engineer is concerned with decomposing the operational behavior of the system, the failure mode analyst is concerned with decomposing the companion fault compensation and mitigation functions. These functions may be embedded with active or passive redundancy, operator procedures, monitoring provisions, alternative operating modes, etc. For example, a top-level function of "Monitor Function" may be decomposed into 5 major subfunctions: "Detect Failure", "Process Detect Status", "Report Status", "Consolidate Status Reports", and "Display Status". Each of these functions is further decomposed in subsequent iterations until their requirements can be allocated to the physical (both hardware and software) system components.

As the baseline configuration is defined, a degree of maturity is reached in which a set of functional block diagrams can be developed to represent the subsystem elements. The functional block diagrams capture the architecture and defined end-item behavior, and compensating provisions. At this point in the development, a Functional Failure Mode Analysis is initiated to verify the requirements derivation and allocation.

6. FMEA TASKS:

A FMEA normally analyzes each single item failure as if it were the only failure within the system. When the item failure is undetectable or latent or the item is redundant, the analysis may need to be extended to determine the effects of another failure, which in combination with the first failure could result in an undesirable failure condition (see 6.5). All single point failures identified during the analysis that result in a non-desirable failure condition must be identified on the FMEA worksheets for proper disposition. Figure 5, extracted from Figure 2, illustrates the sequence of tasks performed during the FMEA procedure. First, the potential item failure modes are postulated; then for each failure mode the frequency of occurrence is assessed, failure consequences are identified and failure modes are grouped into equivalence classes with respect to their failure consequences. The numbers in the boxes refer to the sections in which the indicated task is discussed.

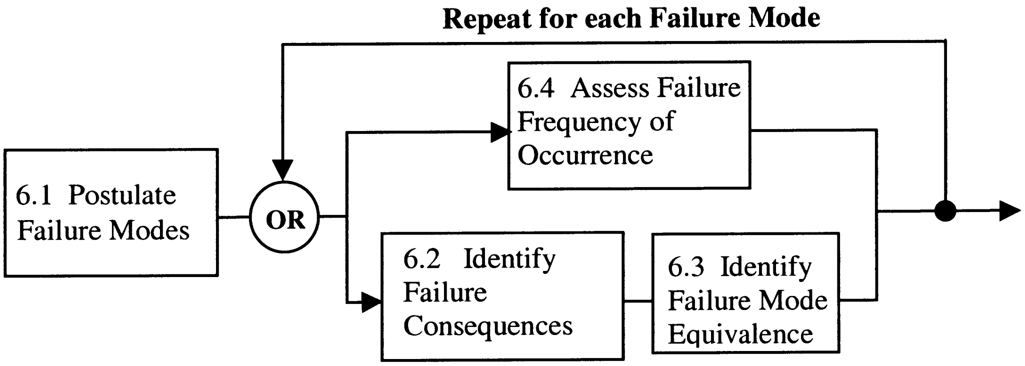

Repeat for each Failure Mode
6.1 Postulate Failure Modes
OR
6.4 Assess Failure Frequency of Occurrence
6.2 Identify Failure Consequences
6.3 Identify Failure Mode Equivalence

FIGURE 5 - FMEA Procedure Tasks

6.1 Postulate Failure Modes:

The analysis begins with a list of potential failure modes. Design modeling begins by specifying the inputs, outputs, functions, operations and structure of the system being developed. The system is then subdivided into smaller subsystems whose inputs, outputs, functions, operations, and structure are specified so as to implement the original system. This conceptual subdividing of the system into ever simpler subsystems often parallels its subdivision into physical hardware modules and software functionality. As a result of the modeling process, systems are described hierarchically by their subsystems, sub-subsystems, software, etc., down to the lowest level components.[4]

During the Functional FMEA, a conceptual model is developed with a functional block diagram of the system, subsystem, product, process, or component being analyzed. The model of the equipment functions consists of partitioning the design in a manner that shows each functional block to have a single functional output. The block diagram indicates the input/output transfer function, the flow of information, energy, force, fluid, etc. within the system, and the primary relationship between the items to be covered in the analysis. Functional failure mode models are assigned to each block resulting in the list of postulated failure modes to be analyzed.

During the Interface FMEA, the functional block diagram is supplemented with unit interconnection diagrams and software interface specifications. The interconnection diagrams and software interface specifications identify the unit interfaces to the extent that fault models postulating the failure modes to analyze can be assigned. The interface failure mode models are assigned to each hardware and software interconnection in the list of postulated failure modes to be analyzed during the hardware and software Interface FMEAs.

[4] The levels in a hierarchical system model are often called "levels of indenture". This terminology is derived from documentation techniques in which the descriptions of subsystems were indented relative to the description of the system of which they were a part. Numbering conventions such as 1.23.245 which identify the system (1), subsystem (23), etc. are often used.

6.1 (Continued):

During the Detailed FMEA the functional block diagram is replaced with the detailed hardware design schematics, or detailed software design documents and source code listings. The schematics identify all piece-part components to the extent that fault models capable of postulating the necessary failure modes to analyze can be assigned. Piece-part failure mode models are assigned to each hardware component resulting in a list of postulated failure modes to be analyzed during the hardware Detailed FMEA. The detailed software design documents and source code listings provide the structure to which software failure modes are assigned for the software Detailed FMEA.

6.1.1 Failure Mode Modeling: System failure mode models are defined from the end-item configuration as reflected in the system drawings and determined by examination of the item inputs and outputs. The postulated failure modes for each function, interface, software element, and hardware piece-part account for the following conditions (Reference 2.3.4):

1. premature operation;

2. failure to operate at prescribed time;

3. failure to cease operation at prescribed time;

4. failure to meet functional specifications:

    a. loss of input/output during operation,
    b. possible modes of internal failure caused by malfunctions and damage from external sources,
    c. degraded input/output or operational capability, and
    d. transient or intermittent operation; and

5. failure conditions caused by the operational and maintenance environment.

Failure mode models define the set of failure modes to be postulated. Failure mode models are defined for the three types of analyses: Functional, Interface, and Detailed.

During the Functional FMEA a failure mode model is developed for each generic function type. Typical failure modes are “function fails to perform” or “function is continuously performed.” Examples of failure modes for the function type heater would be “heater fails to heat” and “heater always heats” (Reference 2.3.10).

6.1.1 (Continued):

Functional FMEA of software is a systematic analysis of the effects of software errors and possibly some hardware failures on system and software behavior. During a software Functional FMEA, potential software errors are viewed in the abstract: as failures of the functions implemented in the software. The intent of the software Functional FMEA is to provide an assessment of the effectiveness of the software architecture and the functional software decomposition as represented by the specific software elements in the design. In most cases, four failure modes are of particular interest when applied to the software elements within the architecture: (1) failure to execute, (2) incomplete execution (e.g., skips and early returns), (3) execution at an incorrect time (early, late, or when it should not have executed), and (4) errors in the software element’s assigned function (incorrect result). For some software, the effects of other failure modes may also need to be assessed. For example, a thorough analysis of an embedded, real-time system will often require a detailed assessment of interrupt timing and priority assignments. Errors in interrupt handling can sometimes lead to deadlocks and race conditions when unexpected execution sequences or rates are encountered. Additionally, assessment of the behavior of the software with certain hardware failures may be appropriate. The analyst should possess a thorough understanding of both the hardware and software elements of a system prior to beginning a software Functional FMEA to ensure that all needed failure modes are considered.

When functional analysis is applied to processes, typical failure mode categories are manufacturing, assembly, receiving inspection, and testing inspection. Process failure modes are described by quantifiable process characteristics that can be corrected. For example, part mis-orientation, part hole off-center, binding, cracked, etc.

During the Interface FMEA a failure mode model is developed for each interface type. Typical failure modes include “signal fails in the open condition”, “signal fails in the short condition”, and “piping fails in the closed position”, etc. An example failure mode for the interface type discrete signal would be “discrete signal fails open” (Reference 2.3.10).

A software Interface FMEA is similar to a Functional FMEA for software, but is intended to focus on failures affecting the interfaces between disparate software and hardware elements. When a software Interface FMEA is done, four failure modes are most often applied to each software interface: (1) failure of the interface to update a value, (2) incomplete update of the interface value, (3) updates to interface values occur at an incorrect time (early, late, or when it should not have executed), and (4) errors in the values or messages provided at the software interface. In some cases, the analyst may also need to evaluate additional failure modes specific to the software or the interface hardware.

During the hardware Detailed FMEA a failure mode model is developed for each component type. Typical failure modes include “stuck high”, “stuck low”, “open”, and “closed”. For example, failure modes normally considered for the component type, capacitor, are “open”, “short”, and “leaking”; for the component type, integrated circuit, they are “output pin stuck high” and “output pin stuck low”; and for the component type, bearing, they are “binding or sticking”, “excessive play”, and “contaminated” (References 2.3.4 and 2.3.10).

6.1.1 (Continued):

A software Detailed FMEA is an extension of the software Functional FMEA applied to the as implemented code. The intent of the analysis is to assess the effect on the software and the system of failures in the software functionality and in the variables used in the software. The analysis is intended as an expansion of the software Functional FMEA for systems that do not have hardware protection for memory, communications and computational hardware. Due to the large size and associated cost of performing a software Detailed FMEA, the analysis is generally reserved for small embedded systems – those systems with minimal or no memory and computational protection provided by the hardware. The failure modes used in the software Detailed FMEA include errors in the implemented software function (e.g., algorithm singularities). Appropriate failure modes for each function must be developed by the analyst based on the specific design being assessed. The effect of incorrect values for each variable must also be examined by assessing the effect of all logical ways that the variable can be incorrect. Table 3 gives failure modes for some common variable types (Reference 2.3.12). As shown for the simple variable types in the table, the software variable failure modes must form a logically complete set of the possible error states for the variable type.

Modern high-level languages, particularly those based on object oriented techniques, include and encourage the use of relatively complex variable types. In some cases, for example, structure variables in ‘C’, determining the underlying variable storage implementation is straightforward. For complex variables in object oriented languages, the analyst may need to develop appropriate failure modes for each variable based on an extensive knowledge of the specifics of the compiler implementation of the language being used for the design. Thus, detailed knowledge of the language specification and of the compiler implementation may be required to allow a software Detailed FMEA to be effectively performed on some complex high-level language designs. Additionally, the analyst will need to ensure that the implementation being analyzed has been limited to well defined language elements – those not left to the compiler developer’s interpretation. If the language use is not limited to well defined elements, the results of a software Detailed FMEA may be incomplete or inaccurate. In some cases, the analyst may need to examine the assembly language level of implementation (or its equivalent) to determine the appropriate failure modes for the variable. Not all compilers will allow that level of detail to be easily accessed.

TABLE 3 - Common Software Variable Failure Modes (Reference 2.3.12)

Variable Type	Failure Modes
Analog (real, integer)	Value exceeds allowed tolerance high.
	Value exceeds allowed tolerance low.
Analog with validity flag	Value is within tolerance; validity flag is set to invalid.
	Value exceeds allowed tolerance high; validity flag is set to valid.
	Value exceeds allowed tolerance low; validity flag is set to valid.
Enumerated (allowed values a, b, c)	Value is set to a when it should be b.
	Value is set to a when it should be c.
	Value is set to b when it should be a.
	Value is set to b when it should be c.
	Value is set to c when it should be a.
	Value is set to c when it should be b.
Enumerated with validity flag	Value is set to a when it should be b; validity flag is set to valid.
	Value is set to a when it should be c; validity flag is set to valid.
	Value is set to b when it should be a; validity flag is set to valid.
	Value is set to b when it should be c; validity flag is set to valid.
	Value is set to c when it should be a; validity flag is set to valid.
	Value is set to c when it should be b; validity flag is set to valid.
	Value is correct; validity flag is set to invalid.
Boolean (true, false)	Value is set to true when it should be false.
	Value is set to false when it should be true.
Boolean with validity flag.	Value is set to true when it should be false; validity flag is set to valid.
	Value is set to false when it should be true; validity flag is set to valid.
	Value is correct; validity flag is set to invalid.

6.1.2 Failure Mode Ratios: The failure mode model includes the fraction of item failures that are in the given failure mode. This allows the item failure probability to be apportioned amongst each of the item failure modes to give the item failure mode frequency of occurrence. Failure mode ratios are best obtained from field data that is representative of the particular item application, but when such data is not available references such as FMD 97 (Reference 2.3.4) can be used. Failure mode ratios for a particular component type may vary depending on the operating environment, manufacturer, application, and other factors.

Typically, item failure modes are considered to be mutually exclusive and the set of failure modes complete; thus the sum of the failure mode ratios, over all the item failure modes, is 1.0. Then when failure rates are used to assess the frequency of occurrence, the failure mode failure rate is the fraction of the item failure rate attributable to the given failure mode. Likewise, if failure probabilities are used the item failure mode probability is the fraction of the failure probability attributable to the given failure mode.

6.1.2 (Continued):

If the failure modes are not mutually exclusive the sum of the failure mode ratios may exceed 1 and the sum of the failure mode probabilities (or failure rates) provides an upper bound on the overall item probability of failure.

6.2 Identify Failure Consequences:

The information required in the FMEA is “What has failed?” (Failure), and “What are the consequences?” (Effect). The attributes of these two requirements are decomposed to derive the set of required FMEA data included in the analysis. Figure 6 illustrates the minimum assessed consequences needed for the FMEA. The numbers in the boxes indicate the section in which that task is discussed.

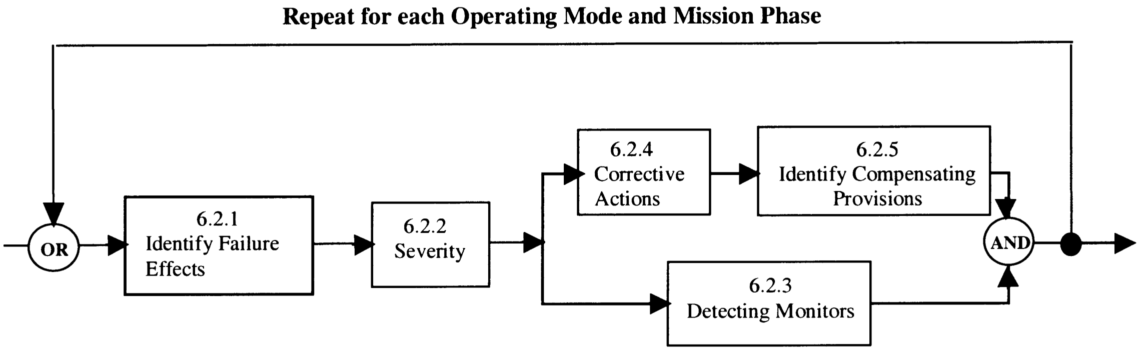

Repeat for each Operating Mode and Mission Phase
OR
6.2.1 Identify Failure Effects
6.2.2 Severity
6.2.4 Corrective Actions
6.2.5 Identify Compensating Provisions
6.2.3 Detecting Monitors
AND

FIGURE 6 - Identification of Failure Consequences

The analysis is conducted for all phases and modes of system operation such as primary and secondary mission objectives, normal operating modes, contingency modes, dormant modes, test modes, ground controlled modes, and autonomous and non-autonomous modes, as defined in the system or end-item requirements documents. Local, next-higher, and end- or system-level failure effects and corrective actions or compensating provisions are analyzed within each applicable operating mode.

Operating modes represent the stimuli and controls applied to the end-item in the form of operational commands, status reports, or tests performed either routinely or on an as-needed basis. The effects of each postulated failure mode, in each operating mode, must be ascertained and recorded, even if it is an undetected, but acceptable, failure. Operating modes are initially identified from a high-level requirements analysis and may be further defined as the design matures. When a failure mode cannot be detected, the analysis defines the requirements for a test.

6.2 (Continued):

Sufficient assessment of all failure mode effects includes identification of the system conditions or operational modes (operational stimuli) that manifest the anomalous behavior. For example, failures within the airplane landing gear that would cause “loss of landing gear extension” will have significantly different impacts when the failure occurs on the ground than if it occurs while attempting to land. Likewise, the discovery mechanism for detecting the failure may be different.

The operational profile is defined in terms of the operating mode in which the system or end-item is functioning. The operational profile includes the sequence of end-item functions necessary for system success, the duty cycle, environmental impacts, and any anticipated maintenance conditions affecting the system/end-item.

6.2.1 Identify Failure Effects: The analysis proceeds by identifying the effects of each postulated failure mode in a bottom up manner beginning with the lowest level items identified. The effects of each failure mode are evaluated with respect to the function of the item being analyzed. Since the item failure under consideration might impact the system at several levels of indenture, the failure effects are then related to the functions at the next higher indenture level of the design, continuing progressively to the top or system-level functions.

Local Effects: The fault description of the local effect(s) gives a detailed accounting, in prose, of the impact the failure has on the local operation or function of the item being analyzed. The fault condition is described in sufficient detail that it can be used with the next-level effects, end-effects, and detecting monitor(s) to identify and isolate the faulty equipment; thus providing a basis for evaluating compensating provisions and recommending corrective actions.

Next-level Effects: Next-level Effects describe the effect the failure has on the next-higher-level operation, function, or status. Descriptions of the next-level effects are normally compiled in a table for consistency of annotation. (See Analysis Libraries in 4.5.)

End-Effects: End-effects describe the effect the failure has on the ability of the end-item to operate or complete the system functions. End-effects also include the effect the failure has on the ability of the end-item to perform its intended mission by providing a “Go/No-Go” assessment of system capabilities. An initial set of end-item, failure effect descriptions is derived from the source requirements and compiled in a table for consistency of annotation. (See Analysis Libraries in 4.5.)

The details of the FMEA analysis are captured on analysis worksheets. The worksheets provide a description of the failure modes and their consequences traceable to diagrams and/or other design documentation. The information captured on the worksheet summarizes the analysis content.

6.2.2 Identify Severity: Severity is an assessment of the significance of a failure mode’s effect on the system, mission, or application. The FMEA ground rules establish the ranking system and appropriate criteria for assessing and classifying the severity of failures for the product being analyzed. The severity classifications and descriptions provide the rules whereby the severity of a particular item failure mode effect can be determined. Severity classifications may change depending on the mission phase or other operational factors that affect utilization of the function.

When items are redundant and there is no detection or no warning that a redundant item has failed, the severity should be assessed as if all of the redundant items have failed.

The number of severity classifications and their descriptions should be tailored to the industry or system being analyzed. Tables 4 through 6 show examples of rankings used in the military, aerospace, and automobile industries. If more than one set of rankings is to be used a cross reference mapping must also be provided so that the scales can be merged.

6.2.3 Detecting Monitors: Proper warnings are necessary to alert the user to unsafe system operating conditions and to ensure satisfactory system status prior to commencing operations. Operational monitoring requirements are derived from the failure effects assessment to support the system mission. Monitors must be strategically located to cover all undetected catastrophic, hazardous, and single-point failures, based on the system requirements and intended use. Once the necessary monitors have been identified a subsequent FMEA iteration is conducted (in support of maintenance activities) to verify that any remaining undetected failure modes comply with the system fault detection requirements. Requirements for fault [detection] monitors are then derived to cover the remaining undetected failure modes. These monitoring requirements include operator procedures and human monitoring, in addition to Built-In Test (BIT). Monitors are located to provide cost effective coverage of the undetected failure modes as measured by the percentage increase per recurring dollar.

A monitor list is created and annotated throughout the analysis to identify the monitors that detect the failure modes being analyzed. Failure modes are assigned to every monitor that provides detection of that failure. The monitor coverage, including all undetected failure modes, can be assessed from this linkage. All monitors which detect a given failure mode must be identified within the analysis. Identification of the detecting monitor is described in the analysis worksheet under "Detecting Monitor". A brief functional narrative is provided describing the monitor(s) that detect the postulated failure mode. The description is predicated on the type of end item under analysis and may include some or all of the following information: monitor type, function or test, measured parameter(s), fault report, or annunciated system indications. Where the monitoring provisions are incomplete, the analysis results will generate requirements and recommend solutions to maximize the fault coverage.

TABLE 4 - 4-Level Military/Government Severity Ranking Criteria (Reference 2.2.2)

Category	Criteria: Severity of Effect
I	Catastrophic: A failure which can cause death or system loss (e.g., aircraft, tank, missile, ship).
II	Critical: A failure which can cause severe injury, major property damage, or major system damage which will result in mission loss.
III	Marginal: A failure which may cause minor injury, minor property damage, or minor system damage which will result in delay or loss of availability or mission degradation.
IV	Minor: A failure not serious enough to cause injury, property damage, or system damage, but which will result in unscheduled maintenance or repair.

TABLE 5 - 3-Level Aerospace Industry Severity Ranking Criteria (Reference 2.1.2)

Category	Criteria: Severity of Effect
Critical	Functions for which the occurrence of any failure condition or design error would prevent the continued safe flight and landing of the aircraft.
Essential	Functions for which the occurrence of any failure condition or design error would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions.
Non-Essential	Functions for which failures or design errors could not significantly degrade aircraft capability or crew ability.

TABLE 6 - 10-Level Automobile Industry Severity Ranking Criteria (Reference 2.1.4)

Effect	Automobile Industry Ranking	Criteria: Severity of Effect
Hazardous (without warning)	10	Very high severity ranking when a potential failure mode affects safe vehicle operation and/or involves noncompliance with government regulations without warning.
Hazardous (with warning)	9	Very high severity ranking when a potential failure mode affects safe vehicle operation and/or involves noncompliance with government regulation with warning.
Very High	8	Vehicle/item inoperable, with loss of primary function.
High	7	Vehicle/item operable, but at reduced level of performance. Customer dissatisfied.
Moderate	6	Vehicle/item operable, but comfort/convenience item(s) inoperable. Customer experiences discomfort.
Low	5	Vehicle/item operable, but comfort/convenience item(s) operable at reduced level of performance. Customer experiences some dissatisfaction.
Very Low	4	Cosmetic defect in finish, fit & finish/squeak & rattle item that does not conform to specifications. Defect noticed by most customers.
Minor	3	Cosmetic defect in finish, fit & finish/squeak or rattle item that does not conform to specifications. Defect noticed by average customer.
Very Minor	2	Cosmetic defect in finish, fit & finish/squeak or rattle item that does not conform to specifications. Defect noticed by discriminating customer.
None	1	No effect.

6.2.3 (Continued):

The FMEA must retain traceability from the failure mode to the system indication. Recording the detecting monitors with the failure modes permits the fault reporting and fault annunciation design to evolve, while minimizing analysis perturbations due to evolutionary design changes. System indications can then be derived from the attributes of the monitors detecting the failure mode. System indications identify what the user sees or perceives and the state of the manifested indication. If there is no system indication of a failure, the analyst must determine if the undetected failure will jeopardize the mission objectives or personnel safety. System indications identify how a failure is discovered and support the maintenance procedures for locating the failed item. Examples of system indications are displayed status messages, visual observations, auditory messages, or user perceived operational malfunctions.

6.2.4 Corrective Action Recommendations: Corrective actions are needed for faults having significant consequences – for example, unsafe conditions, critical (mission, safety) single point failures, adverse effects on operating capability, high maintenance costs, and undetectable (hidden or dormant) faults. They may not be needed if the risks for the specific consequence(s) of the failure are acceptable based on a low enough probability of occurrence. Corrective actions generally take the form of changes in requirements, design, processes, procedures, or materials to eliminate a design deficiency.

Development of an appropriate corrective action usually requires understanding and eliminating the root cause of the failure mode. Special attention to the failure mode causes may be needed to ensure that proper materials are used when the operational environment is especially severe due to effects such as extreme temperature cycling, very high or very low operating temperatures, the presence of corrosive chemicals, etc. Careful analysis of the causes may suggest ways to eliminate the failure mode. For example, a metal valve used to control the flow of a gas might fail to close properly due to corrosion. The corrosion might be caused by electrolysis between different metals or by a reaction of the valve material with the gas. Careful consideration of the valve material could eliminate this failure mode or indicate the need for a maintenance inspection/ replacement requirement.

Some examples of failure causes are:

1. incorrect material specification;
2. overstressing of a component;
3. insufficient lubrication;
4. inadequate maintenance instructions;
5. poor protection from the environment;
6. incorrect algorithm; and
7. software design error, including software requirements errors.

Once a corrective action is implemented and validated the affected fault analyses are revised to reflect the new baseline configuration.

6.2.5 Identify Compensating Provisions: When design changes to correct a deficiency are not possible or feasible, compensating provisions must be identified to eliminate, circumvent or mitigate the effect of the failure when it occurs. Such provisions are often in the form of design provisions or designated operator actions that allow continued safe operation when a failure occurs.

Some examples of design compensating provisions are:

1. addition of redundant items that allow continued safe operation in the event of a failure;
2. requirements for safety or relief devices , monitors, or alarms that limit damage caused by a failure; and
3. requirements for alternative modes of operation such as backup or standby items.

Such changes become part of the baseline design and the affected fault analyses are revised.

Operator actions that can circumvent the effects of a failure also serve as compensating provisions. The consequences of any probable incorrect actions(s) by the operator in response to an anomaly also must be considered and recorded in the analysis. Examples of operator compensating actions include:

1. execution of system commands and controls;
2. switching to alternative modes of operation, possibly using backup or standby items; and
3. conducting operational procedures.

6.3 Identify Failure Mode Equivalence:

Identification of fault equivalence groups permits the analyst to manage failure consequences instead of individual failure modes (Reference 2.3.11). This technique reduces the magnitude of the analysis effort while improving its consistency. The common element between each analysis type is the subsystem fault, identified by a Fault equivalence Identification Number (FIN), containing a description of the fault’s local effect(s), next-level effect(s), end-effect(s), severity, compensating provisions, and detecting monitor(s). Those failure modes that exhibit identical consequences are termed “fault equivalent failure modes” and grouped by the same fault equivalence number. Failure modes whose consequences are not identical to previously analyzed faults are assigned new FINs.

For example, in Figure 7a the EMI Input Filter has the two functional failure modes “does not Filter EMI” and “no output” which are assigned FINs 001 and 002 respectively in Figure 7b. At the detailed level shown in Figure 7c, the piece-part failure modes “resistor short”, “inductor short”, and “capacitor open” cause the device not to filter the input and thus have the same failure consequence as the functional failure mode "does not filter"; hence, they are assigned FIN 001. Similarly, the piece-part failure mode “resistor open” and “inductor open” have the same consequences as the functional failure mode “no output”and are assigned FIN 002. However, the piece-part failure mode “capacitor short” results in a new failure consequence, “shorted output”, and must be assigned a new FIN, 003. At this point in the analysis, it is useful to add the functional failure mode “shorted output” to the functional analysis and to rename the functional failure mode “no output” as “open output” to distinguish it from the newly added “shorted output” failure mode.

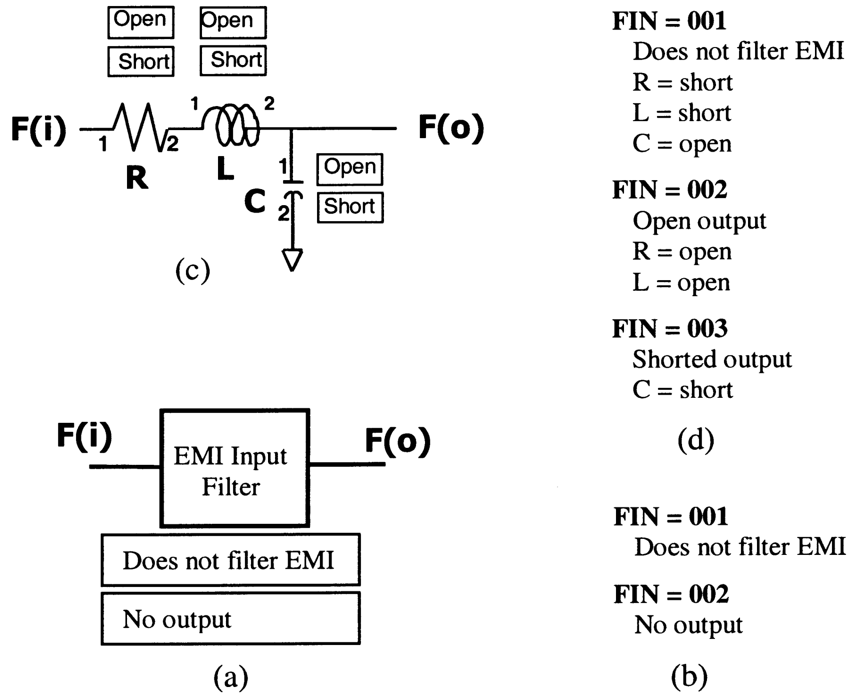


FIGURE 7 - Example Illustrating Failure Mode Equivalence

6.3 (Continued):

The fault equivalence number represents the set of consequences common to all the failure modes in the group identified by the FIN. The FIN is generated on a functional basis and thus allows previously generated fault information to be traced and used in subsequent analysis. For example when the effects of an interface failure mode are identical to those of a previously analyzed functional failure mode the interface failure mode is assigned the FIN for the already generated and recorded functional fault, thereby maintaining continuity. Similarly, in the Detailed Analysis piece part failure modes that result in consequences identical to those of previously analyzed functional and interface failure modes are assigned to the already generated and recorded fault group via the FIN. As illustrated in Figure 8, the fault equivalence group can be thought of as a bucket defined by identical consequences, that holds failure modes from the functional, interface, and detailed analyses.

The use of fault equivalent numbers for group failure modes with identical consequences facilitates integrating the Functional FMEA with the subsequent Interface and Detailed FMEAs and supports timely feedback to the designer. Each group is organized within the FMEA worksheets by sorting on the FIN.

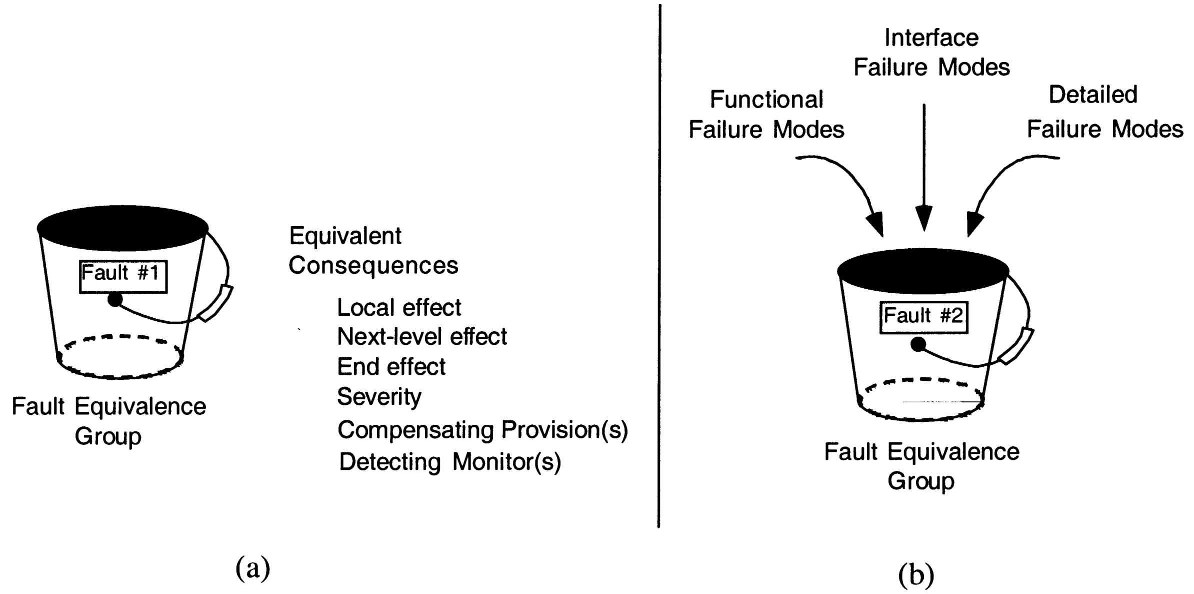


FIGURE 8 - Fault Equivalence Grouping

6.4 Assess Failure Frequency of Occurrence:

The failure frequency of occurrence quantifies the expectation that an item fails in the given failure mode. The failure mode frequency of occurrence can be estimated from field data, laboratory or simulation studies, experience with similar systems, tables of generic component failure rates, analysis of the item design and other methods.

In practice the item failure probability is most often expressed by giving a corresponding failure rate. (See the discussion of constant failure rates in 6.4.1.) Failure rates are best obtained from field data, but when field data is unavailable, references such as Non-Electronic Parts Reliability Data (Reference 2.3.3), Reliability Prediction of Electronic Equipment (Reference 2.2.3), or Reliability Prediction Procedure for Electronic Equipment (Reference 2.3.8) provide a standardized approach for predicting the rates of piece-part failures. If sufficient data is not available, a subjective assessment of item failure mode probabilities can be made, based on historical data from similar systems or processes. However, these “judgments” should be updated as data becomes available from simulations, prototype evaluation, or early production units.

The hardware failure frequency is specified as a probability over a stated performance period of time t. The probability is computed with respect to an operational interval based on the mission profile for the product being analyzed. The failure probability is based on field data modeled by an appropriate probability distribution function; the exponential and Weibull distributions are the most often used. When the exponential distribution is assumed a constant failure rate, λ, applies.

6.4 (Continued):

Depending on the product, the operational period may be measured in units of time (e.g., operating time, warranty period, mission time, 1 year, 5 years, etc.) or number of events (e.g., landings, operating procedures, items placed, holes drilled, opportunities for error, etc.).

For a Functional FMEA, the failure rates attributable to a function can be used to calculate the function failure probability. During Interface FMEA, the piece-part failure rates attributable to an interface can be used to calculate the interface hardware failure probability.

Software failure rates and the associated rates for given software failure modes are not standardized throughout industry, and may in fact be largely application and developing organization dependent. The analyst will have to estimate failure and failure mode probabilities based on historical records for software programs within the specific industry (or even the development group) for the product being analyzed. Some help in software failure rates applicable to large aerospace command and control systems is provided in Reference 2.3.13.

For software failure rates, the analyst may need to depend on experience, both personal and recorded experience within the industry producing the product type and the company producing the specific product being analyzed, to estimate failure rate ranges from tables similar to Table 8. These estimates should be updated once the product enters software testing and again during field deployment based on the rate of defect discovery and the relative strength of the testing performed. It may be possible to at least partially estimate test and environment exposure strength with respect to specific software failure modes based on the work of Friedman and Voas (Reference 2.3.14) in the area of software testability.

6.4.1 Constant Failure Rate: Typically, electronic hardware failure mode frequencies of occurrence assume a constant item failure rate over much of the product lifetime. For the particular case of a constant failure rate λ, and an operating time t, the failure mode probability is derived from the exponential distribution:

$$\Pr(\text{failure mode}) = 1 - e^{-\lambda t} \qquad \text{(Eq. 1)}$$

Here t is the time period of interest (normally the mission time in hours) and lambda (λ) is the failure mode failure rate (generally in failures/hour). Lambda is calculated as the product of the failure mode ratio and the item failure rate:

$$\lambda = \alpha\lambda_p \qquad \text{(Eq. 2)}$$

where:

α = Failure mode ratio
λ_p = Item failure rate

Failure mode ratio (α): The failure mode ratio is the fraction of item failures apportioned to the failure mode under consideration.

6.4.1 (Continued):

Item failure rate (λ_p): The item failure rate (λ_p) is obtained from field experience data or an appropriate reliability prediction. The failure rate units used should be compatible with those used for the operating time.

Operating time (t): The operating time is expressed in hours, number of events, number of operating cycles, or other units appropriate to the item and mission. In Equation 1 t may also represent the number of events, in which case λ must be in terms of failures/event.

6.4.2 Process Variation: The probability that an item parameter will be out of specification due to variations in the manufacturing process is given by the Process Capability Index, C_{Pk}. The C_{Pk} is estimated from the specified component tolerance and the expected variability of the manufacturing process measured at the 3σ level. (This is the range covered by ±3 standard deviations of the parameter value from its mean; it encompasses approximately 99.7% of the items assuming a normal distribution.)

The C_{Pk} index is defined as the ratio of the difference between the process mean parameter value ($\bar{x}$) and the nearest tolerance limit to the 3σ process variation (Reference 2.3.16.):

$$C_{Pk} = \frac{\min(|T_U - \bar{x}|, |T_L - \bar{x}|)}{3\sigma} \qquad \text{(Eq. 3)}$$

where:

T_U = Upper tolerance limit
T_L = Lower tolerance limit

For a symmetric process, a C_{Pk} of 1 implies that the tolerance limits are $\pm 3\sigma$ of the production process.

Table 7 gives the probability that the production process produces an item that is out of specification for various values of C_{Pk}. Since C_{Pk} considers the nearest tolerance limit to the mean, the probability out-of-tolerance in Table 7 is a worst case estimate.

TABLE 7 - C_{Pk} Index Probability Values

C_{Pk}	Tolerance Limit	Probability Out of Tolerance	Defect Rate*
0.33	1σ	**0.3173**	317,300 PPM
0.67	2σ	**0.045503**	**45,503** PPM
1.00	3σ	**0.0027**	2700 PPM
1.33	4σ	**0.0000635**	63.5 PPM
1.67	5σ	**0.000000573**	0.573 PPM
2.00	6σ	1.973×10^{-9}	1.97 PPB
2.33	7σ	**2.560×10^{-12}**	**2.56 PPT**

* PPM = Parts Per Million; PPB = Parts Per Billion; PPT = Parts Per Trillion

6.4.2 (Continued):

Usually $\bar{x}$ is not known until the production process is set up and some units have been produced. Thus, early in the design process the C_P index, defined as:

$$C_P = \frac{T_U - T_L}{6\sigma} = \frac{\Delta T}{3\sigma} \qquad \text{(Eq. 4)}$$

may provide an estimate of the out-of-tolerance probability. The C_P and C_{Pk} indices are related by the factor K:

$$C_{pk} = (1 - K)C_p \qquad \text{(Eq. 5)}$$

where:

$$K = \frac{|\bar{x} - T_n|}{(T_U - T_L)/2} \qquad \text{(Eq. 6)}$$

and T_n is the nominal parameter value.

Table 7 applies to the C_P index as well as the C_{Pk} index. A C_{Pk} (or C_P) of 1.33 or higher is generally considered “good”, 1.0 to 1.32 “marginal”, and 1.0 or less “bad”.[5]

[5] As process technology improves, these definitions of "good", "marginal", and "bad" will likely change and higher C_P or C_{Pk} indices will be expected.

6.4.3 Qualitative Assessment: When specific parts, configurations, or failure rate data are not available the probabilities of occurrence for the failure modes identified in the FMEA may be assessed qualitatively. Individual failure mode probabilities of occurrence should be grouped into distinct, logically defined levels, which establish the qualitative failure probability level for entry into the appropriate worksheet column.

The failure mode probability of occurrence is a numeric quantity but for convenience it may be described qualitatively in linguistic terms or as an interval. In either case a general definition of the terms or ranges must be provided in terms of numerical failure probabilities. An example of such a labeling used in the automobile industry is given in Table 8.

TABLE 8 - Qualitative Probability Intervals (Reference 2.1.4)

Probability of Failure	Probability	Ranking
Very High: Failure is almost inevitable	≥0.5, (greater than 1 in 2)	10
	0.33, (1 in 3)	9
High: Repeated failures	0.125, (1 in 8)	8
	0.05, (1 in 20)	7
Moderate: Occasional Failures	0.0125 (1 in 80)	6
	0.0025 (1 in 400)	5
	0.0005 (1 in 2000)	4
Low: Relatively few failures	0.0000667 (1 in 15,000)	3
	0.00000667, (1 in 150,000)	2
Remote: Failure is unlikely	≤0.000000667, (1 in 1,500,000)	1

6.5 Failure Latency Analysis:

Failure modes of items for which the operator cannot be made aware of the failure are called undetectable faults. Failure modes for which the annunciation to an operator is delayed beyond some acceptable threshold such that their occurrence cannot be effectively circumvented or corrected, are called dormant or latent faults. Where a single item failure is undetectable or latent, the FMEA may need to be extended to determine the effects of other failures in combination with the first undetectable or latent failure. The need for such an analysis is identified in the analysis plan. A combinatorial analysis that evaluates various combinations of faults is used to determine the occurrence probabilities of any potential catastrophic or hazardous failure conditions.

A fault tree analysis is one widely used type of combinatorial analysis that is particularly well suited for handling combinations of multiple failures (Reference 2.2.5).

7. FMEA DOCUMENTATION AND REPORTING:

The FMEA documentation is the finished record of the analysis. It should be clear, comprehensive, and sufficiently organized to allow for an independent review and authentication of the analysis. The results should be formally documented and summarized at the appropriate design reviews. FMEA worksheets should be documented using electronic media and maintained in industry standard databases to take advantage of computer assisted processing. This allows for computerized grouping, sorting and graphical representations that facilitate technical reviews.

The primary content of the FMEA report includes:

1. a description of the system or end-item being analyzed;
2. block diagrams;
3. FMEA ground rules and assumptions;
4. the analysis worksheets for each item; and
5. a summary of the analysis results.

7.1 System or End-Item Description:

The system or end-item description includes:

1. the identifying name of the system or end-item;

2. identification of all items constituting the system or end-item down to the lowest level of indenture analyzed;

3. a functional description of the system or end-item;

4. functional descriptions of all items analyzed; and

5. a description of each mission and mission phase which identifies the tasks to be performed and their operating modes.

7.2 Block Diagrams:

Block Diagrams are included in the report to identify all the elements (normally at the LRU level) required for operation of the system or end-item. The diagrams include any alternate paths introduced by redundancy, interrelationships, and interdependencies of the functions. The elements on the diagram are usually identified by a reference number that is also used as the item identifier on the FMEA worksheet.

7.3 FMEA Ground Rules and Assumptions:

The report includes a list of the ground rules and assumptions that form the basis of the FMEA. The list includes those ground rules and assumptions identified during the FMEA planning phase (refer to 4.1) and also any changes or new ground rules or assumptions identified while developing the FMEA.

7.4 Analysis Worksheets:

The details of the FMEA for each item are captured on the analysis worksheets. The worksheets provide a description of the failure modes and their consequences traceable to diagrams and other program documentation. The information captured in the worksheets summarizes the analysis content.

The integrator who must integrate all the subsystem analyses into an overall system FMEA may require suppliers to standardize their worksheet data presentation and use a common format. Thus, it is important to obtain concurrence from the customer and integrating contractor on what data is to be contained within the worksheet. If a specific data format is not prescribed, one that recognizes the design complexity and unique applications of the equipment being analyzed should be used.

Generally the worksheets correlate equivalent failure modes to ensure a consistent and accurate analysis review. The following data items are covered in the FMEA worksheet. Figure 9 illustrates the interrelationships of these data items.

7.4.1 Version/Date: The date the FMEA was prepared or last updated and the revision level are included on all forms to identify where updates have been made. Revision status is required to correlate the analysis with the design baseline.

7.4.2 Analyst: The analyst is the person responsible for assessing the failure consequences shown on the worksheet.

7.4.3 End-item/Process Identifier: The end-item or process name identifies the configuration for which the failure modes are analyzed. The identifier must be consistent with those used for the block diagrams, schematics, and other drawings to ensure FMEA traceability.

7.4.4 Subsystem/Subprocess Identifier: The subsystem or subprocess name of each indenture level above the function or item being analyzed is listed so that items and equipment in the system can be unambiguously identified. The identifier must be consistent with those used for the block diagrams, schematics, and other drawings to ensure analysis traceability.

7.4.5 Item/Function/Action Name: The name of the item, function, interface, piece-part or process action being analyzed is listed. The name must be consistent with those used for the block diagrams, schematics, and other drawings to ensure traceability.

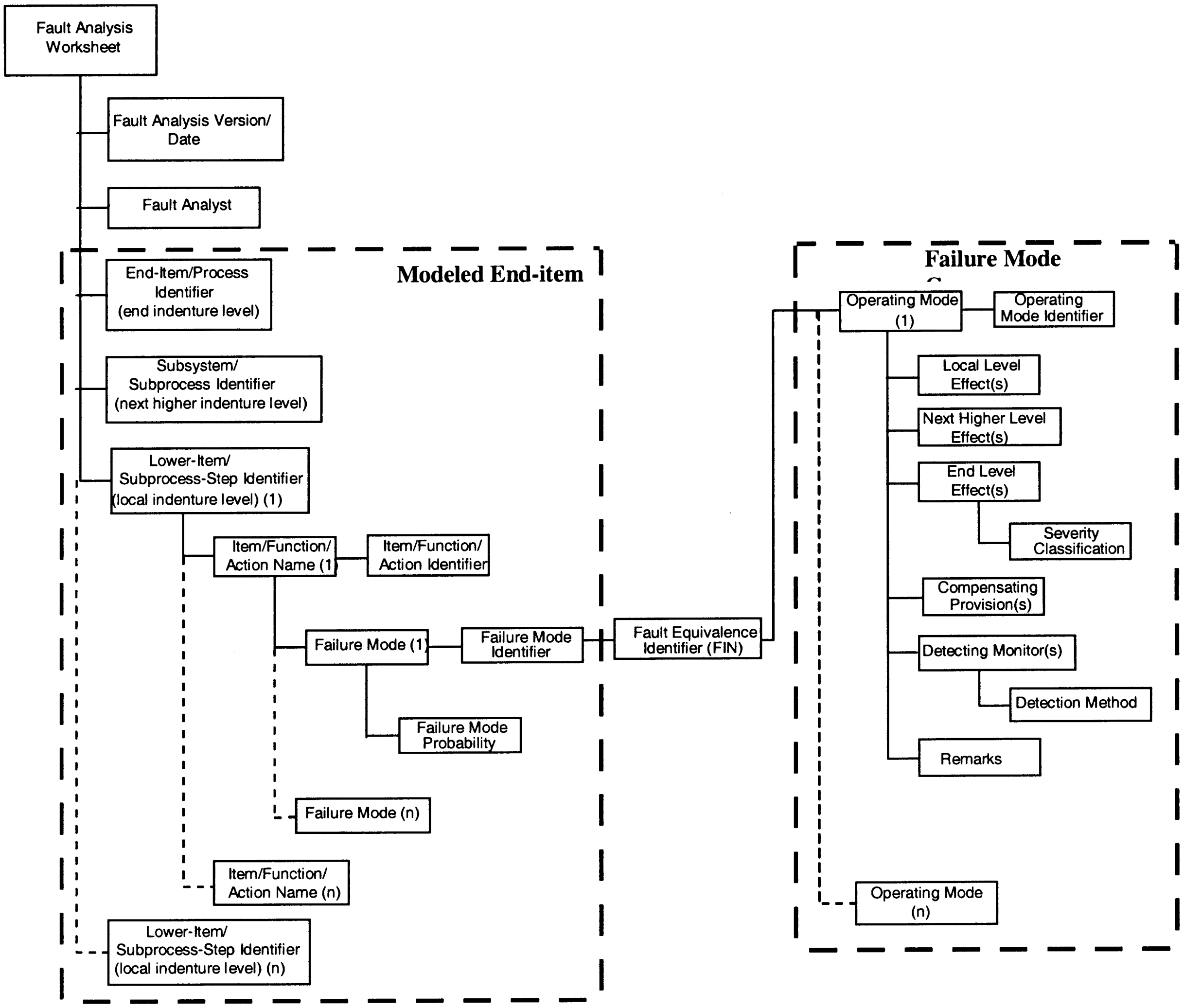

Fault Analysis Worksheet
Fault Analysis Version/ Date
Fault Analyst
Modeled End-item
End-Item/Process Identifier (end indenture level)
Subsystem/ Subprocess Identifier (next higher indenture level)
Lower-Item/ Subprocess-Step Identifier (local indenture level) (1)
Item/Function/ Action Name (1)
Item/Function/ Action Identifier
Failure Mode (1)
Failure Mode Identifier
Failure Mode Probability
Failure Mode (n)
Item/Function/ Action Name (n)
Lower-Item/ Subprocess-Step Identifier (local indenture level) (n)
Fault Equivalence Identifier (FIN)
Failure Mode
Operating Mode (1)
Operating Mode Identifier
Local Level Effect(s)
Next Higher Level Effect(s)
End Level Effect(s)
Severity Classification
Compensating Provision(s)
Detecting Monitor(s)
Detection Method
Remarks
Operating Mode (n)

FIGURE 9 - FMEA Data Item Relationships

7.4.6 Item/Function/Action Identifier: A unique identifier is assigned to each item, function, interface, piece-part or process-action being analyzed. The function number is annotated to the diagrams for analysis traceability. When combined with the subsystem indenture, the identifier allows each element to be uniquely identified.

7.4.7 Failure Mode: The manner or mode in which the item, function or process-action being analyzed fails is listed. Similar failure modes should be described in the same way to ensure consistency in the analysis.

7.4.8 Failure Mode Identifier: A unique identifier is assigned to each unique item, function or process-action failure mode being analyzed.

7.4.9 Failure Mode Probability: Each failure mode should include an estimate of its probability of occurrence.

7.4.10 Fault Equivalence Identifier: A unique identifier is assigned to all postulated failure modes whose consequences (local level effects, next higher level effects, end level effects, severity classification and detecting monitor) are identical for like operating modes.

7.4.11 Operating Mode(s): The FMEA includes a concise statement of the operational state or mission phase in which the failure is analyzed.

7.4.12 Operating Mode Identifier: A unique identifier is assigned to each operating mode. Within each identified operating mode, the following consequences are assessed:

7.4.12.1 Local Effect(s): The FMEA includes a brief narrative description of the impact the postulated failure mode has on the item, function or process task in the local indenture level under analysis. Local failure effects from failure modes that exhibit identical consequences should be described in the same way to ensure consistency in the analysis.

7.4.12.2 Next-level Effect(s): The FMEA includes a brief narrative description of the impact the postulated failure mode has on the operation and function of the next higher indenture level above the functioning entity within the system. Next-level failure effects from failure modes that exhibit identical consequences should be described in the same way to ensure consistency in the analysis. In some cases a supplier might not know the next-level effects for the items that they supply; it then becomes the responsibility of the system owner to ensure that those effects are correctly assessed.

7.4.12.3 End-level Effect(s): The FMEA includes a brief narrative description of the total effect the postulated failure has on the operation, function or status of the system or end-item at the upper-most indenture level. End-effects provide an orderly and thorough evaluation of the effects of foreseeable failures and other events on the system operation and safety. End-level failure effects from failure modes that exhibit identical consequences should be described in the same way to ensure consistency in the analysis. In some cases a supplier might not know the next-level effects for the items that they supply; it then becomes the responsibility of the system owner to ensure that those effects are correctly assessed.

7.4.12.4 Severity: A severity classification category is assigned to each failure mode according to the failure effect(s). Generally the effects of the failure are assessed at the system level.

7.4.12.5 Compensating Provision(s): The FMEA identifies the design provisions or operator actions that will eliminate, circumvent, or mitigate the effect of the failure.

7.4.12.6 Detecting Monitor: The FMEA includes the identification of the monitor(s) that are used to detect the failure modes when they occur.

7.4.12.7 Detection Method: A description of the methods(s) by which the failure mode is perceived by operators or maintenance personnel is listed.

7.4.13 Remarks: Any remarks pertaining to and clarifying the data in the worksheet or recommendations for design improvements.

7.5 Summarize Analysis Results:

The analysis summary is compiled from information extracted from the analysis worksheets, key findings, design solutions and reports on any existing design flaws found in the analysis. The analysis summary should include:

1. A cross reference to the analysis worksheets listing the functions, piece-parts, and/or process tasks analyzed;

2. Any items omitted from the analysis along with the rationale for their exclusion;

3. A summary of the major failure effects compiled and extracted from the analysis worksheets including the number of failure modes which lead to a given effect, and the probability (or failure rate if applicable) for each major effect. Where possible, the summary report should estimate the probability of any failure condition that would prevent continued safe operation of the system. The summary is grouped by severity classification (from highest to lowest severity) with each failure mode in order of decreasing probability of failure (or failure rate). For a large system the summary may be limited to only the worst severity (safety related) and highest probability items.

4. Identification of all single failure points along with their end level effect and severity classifications. (This confirms the adequacy of fail-safe design features or highlights the need for redundancy or backup functions to the designer). The summary is grouped by severity classification (from highest to lowest severity) with each failure mode in order of decreasing probability of failure (or failure rate);

5. A list of all failures and their probabilities of occurrence (if available) causing system level effects with a severity above a predetermined level determined by the program. The summary is grouped by severity classification (from highest to lowest severity) with each failure mode in order of decreasing probability of failure (or failure rate);

6. When undetected failures cannot be eliminated, they must be specifically identified, their probability of failure (or failure rate) identified and their significance dispositioned;

7. A list and count of all latent failures and what the detection method is for each latent failure along with the operating modes in which they are undetectable;

7.5 (Continued):

8. Probability of failure (or failure rate) totals by LRU or subsystem analyzed; and

9. A list of failure modes by detection method.

In addition, summarized reports may be needed to support specialized programs. For example items 3, 6, and 7 support a safety assessment of multiple failures such as a fault tree analysis (Reference 2.2.5), or a Markov analysis.

PREPARED UNDER THE JURISDICTION OF
SAE SUBCOMMITTEE G-11R, RELIABILITY COMMITTEE OF
G-11, RELIABILITY, MAINTAINABILITY, AND SUPPORTABILITY (RMS) DIVISION