sales@missionreadysoftware.com
321-514-4659
http://missionreadysoftware.com

Software Failure Modes Effects Analysis Services

Value statement

When conducted properly and early in development, the software Failure Modes Effects Analysis can identify defects that are typically overlooked in development. An "edge case" is an operation that is guaranteed to execute that no one thought about beforehand. Think of the Software FMEA as the edge case revealer. The types of defects identified by the FMEA are those that are very unlikely to be found in testing largely due to the fact that software is tested to show requirements and not to find failures. Even if the defects uncovered by the software FMEA are found in testing, they will be expensive to fix because they affect specifications and design.These are the reasons for conducting a software FMEA:

These are typical goals for SFMEA
To provide for fault tolerance in areas of high risk
To ensure that the failure modes are explicitly covered in the test plan, simulator and BIT
To strengthen the requirements to include off nominal paths and states
To strengthen the use cases to address off nominal cases
To identify failure modes that are difficult to detect in testing
To ensure that the software is user friendly
To identify information appropriate for a service or user manual (i.e. common workarounds, etc.)
To strengthen the interface design
To strengthen the detailed design (algorithms, logic, data)
To identify items applicable for a health monitoring software (HMS)

The world's largest database of software failures

Mission Ready Software has analyzed almost 1 million software failures. From this, we authored the Common Defect Enumeration, which is currently published on the DAU R&M CoP website. The origin of the root causes is shown below.

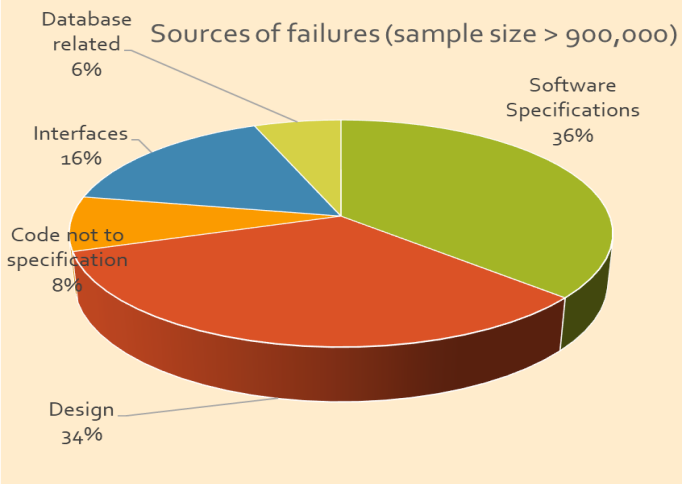

Figure 1 – Origin of software failures

- Contrary to popular belief 70% of them originate in the specification and design
- Defects that originate in the code are less expensive to fix than defects that originate in the specifications and design

The Common Defect Enumeration is the basis for the software FMEA. We focus on the failure modes and root causes that have affected mission critical systems since 1962. Mission Ready Software has been conducting software FMEAs since the 1980s on mission and safety-critical software. Our team of trained analysts know what works and what doesn't with regards to getting the most value from this analysis with the least number of resources. Software FMEAs **aren't** effective if:

- They take too long to finish and therefore cannot affect design
- They are done after all of the code is written or tested
- They are done without inputs from a cross functional team
- They take the "black box" which assumes software components fail like hardware components
- They don't focus on software centric failure modes such as those in the Common Defect Enumeration
- They assume the source of a failure is due to exactly one line of code
- They don't involve at least one individual who understands the software
- The analysts conduct the FMEA against subject matter expert opinion about what the software should do versus what the software is specified and designed to do
- They assume that the software requirements are correct
- They use "eyeball" methods of assessing likelihood
- There are no recommended changes to the specifications, design, test procedures to control the failure modes

Our SFMEA process complies with:

- SAE ARP 5580
- IEEE 1633
- Mil-std-882E/Joint Software Systems Safety Engineering Handbook/AOP-52
- SAE-JA-1003
- SAE-1025 (WIP)
- IEC60812
- AIAG-VDA

Approach

The below is a typical statement of work which will be tailored to your application, the current state of the software (in requirements, design, code, test, etc.), time and resource limitations.

MISSION READY SOFTWARE
sales@missionreadysoftware.com
321-514-4659
http://missionreadysoftware.com

Plan the SFMEA

The below factors determine how the SFMEA is applied to a software system under analysis. Each of these will be considered when constructing the SFMEA.

- The software product can be viewed from several viewpoints – top level, capability level, requirements level and interface level. Each of these can introduce defects that cause failures. Those defects are itemized in the Common Defect Enumeration.
- Every software application has a unique set of risks that drive the tailoring of the viewpoints and the failure mode selection. Of the 400+ failure mode/root cause pairs that apply to nearly all software systems, typically only a few dozen are likely for a particular software system. Software FMEAs are most effective when a root cause analysis is performed up front to identify the most common failure modes and root causes.
- Every software FMEA has a unique set of goals. Those goals determine which viewpoints to select and what the finished SFMEA will be used for. Typical goals are to strengthen the requirements and test plans.
- Today's software systems are too large to analyze exhaustively– intelligent risk-based tailoring of the requirements, design, and code is necessary. The functions and features with the most risk are analyzed first.

Analyze the failure modes

The functional FMEA focuses on what the software does per the specifications and design. The FMEA can be conducted on an entire system or a capability within that system or the interfaces within the software. There are 163 total CDEs; however, for any given system or capability, several of the CDEs aren't applicable. For example, not every application has machine learning or a user interface.

Table 5 – Failure modes and root causes

Failure mode	# of root causes
State management - The software is unable to maintain state, executes incorrect transitions, dead states, etc.	15
Error handling - The software is unable to identify, and handle known system faults in hardware, communications, computers, software components, etc.	40
Faulty functionality – The software does the wrong thing	9
Processing- The software cannot handle peak loading, extended duration, file I/O etc.	9
Faulty sequencing – The software executes out of order	9
Faulty timing – The software executes in the correct order but wrong time	19
User is forced/allowed into an error by software	11
Faulty data definition – conflicts between units, data size, etc.	21
Faulty algorithm – Wrong algorithm or correct algorithm implemented incorrectly	11
Faulty machine learning – Faulty labeling, data collection, modeling	18

sales@missionreadysoftware.com
321-514-4659
http://missionreadysoftware.com

Our assessors use the Requs Software FMEA to expedite the analysis. The assessor describes the system and software and it determines which of the CDEs are applicable simply based on the inventory of capabilities.

Next the assessors complete the failure mode analysis section. The software determines the relative likelihood of each CDE causing a failure based on development and test practices that are planned for the software and our database of failures. Then, your assessor will meet with at least one subject matter expert from your organization who will confirm the effects and severity.

Identify Mitigations and Track to Closure

The Critical Items List is generated once the severity assessment is complete. The Requs Software FMEA software provides recommended mitigations for the critical items for changing the requirements or use cases and developing tests. The assessor will augment these as needed and prepare the team. The development and test team review and modify as needed. From that point onward the customer tracks the items to closure.

Related products and training

Related products	Related training
Requs Software FMEA	**https://missionreadysoftware.com/software-fmea-training-2/**

Pricing

Call 321-514-4659 or email sales@missionreadysoftware.com for quotation.