Using FMEA to Improve Software Reliability

Kraig Strong

Kraig.Strong@Tektronix.com

Abstract

Failure Mode and Effect Analysis (FMEA) is a methodology widely used by hardware designers to model and avoid field failures. Increasingly, this methodology is being adapted to modeling software systems for improving reliability. Among the methods, tools, and practices for improving software reliability, FMEA is arguably the least costly, easiest to learn, and most effective.

The presentation will review the essential elements of the FMEA methodology as it is applied to systems that involve software components. These essential elements include the Functional Block Diagram (FBD) and the FMEA worksheet. The FMEA worksheet is the final work product in the process and culminates in a prioritized list of recommended actions.

Examples, lessons learned, and recommendations will be offered to reinforce the FMEA concepts and provide helpful guidance for those interested in applying FMEA on their software projects.

Biography

Kraig Strong is a software quality engineer at Tektronix in Beaverton, Oregon. Prior to joining Tektronix, he spent a few years as a manufacturing test engineer where he used FMEA on a daily basis. Kraig is bringing over his quality-focused background in order to help deliver more reliable software and products at Tektronix.

Kraig has a B.S. in Electrical and Computer Engineering from Oregon State University and is currently pursuing an M.S. in Computer Science from Portland State University.

1. Introduction

Failure Mode and Effects Analysis (FMEA) is a methodology to find potential failures before they occur. While FMEA identifies individual failure modes, its primary benefit is the early identification of system failure modes so a solution can be designed to mitigate the potential failure. It is a methodology to design reliability into a system. In a FMEA, numerical weights can be applied to the likelihoods of each failure, as well as the severity of the consequences. FMEA is a very cost-effective, easy to learn, and productive way to design a more reliable system.

Although this method was not originally created for software systems, we can translate the principles over to software and take advantage of the many benefits that FMEA has to offer. These benefits include:

- Facilitates early identification of failure points and system interface problems
- Yields a better understanding of planning/scheduling by revealing additional work efforts
- Enables early test planning
- Provides a single worksheet summary of analysis
- Requires minimal training to participate, or even lead a FMEA
- Increases reliability in products

The outcome of this process is a single FMEA worksheet containing a list of failure modes prioritized by risk in the system under study. This prioritized list identifies the most valuable places to focus extra design and test efforts to produce a more reliable system.

Section 2 of this paper presents the key elements of FMEA. Section 3 is an introduction to a Manufacturing Process FMEA for those that are unfamiliar with basic FMEA principles. Section 4 relates to FMEA for software systems.

2. Key Elements of FMEA

A typical FMEA is a team activity, accomplished in one or more meetings. The owner of the system/sub-being analyzed, project leads, QA, and at least one domain expert typically attend the meeting(s). Anybody with even moderate knowledge of the system can also attend and be of great benefit to the outcome. During the meeting, these seven essential steps provide guidance through the process:

1. Define Failure Modes – *What can go wrong here?*
2. Define Effects – *What will happen then?*
3. Describe Targets – *Who will suffer from the failure?*
4. Find Root Causes – *Why will that happen?*
5. Prioritize the Risks – *What is the likelihood?*
6. Define Solution Actions – *How can this be prevented?*
7. Define Current Prevention and Detection Methods – *What is currently being done?*

These steps are repeated throughout the meeting, resulting in the FMEA worksheet with a prioritized list of risks and their associated failures in the system.

3. Manufacturing Process FMEA

There are three main types of FMEA:

- Process: Used to analyze manufacturing and assembly processes. This is arguably the most straightforward, and easiest to understand type.
- Hardware: Used to analyze hardware systems both before (concept design) and after (detailed design) the hardware design is completed.
- Functional: Based on a functional breakdown of a system. Used to analyze high-level functional blocks of a system. This is what is used for software evaluation. This paper will use the term 'Software FMEA' to represent this type.

Since the Process FMEA is the most straightforward, an example will be provided to clarify the concepts and steps to perform a successful FMEA. Section 4 will provide a software FMEA example. The example used below is a simple process of attaching a printed circuit board (PCB) to a heat sink to sheet metal using thermal tape. This is done to help the PCB stay cool to increase the lifespan of the components.

Before jumping into the steps, first define what is to be analyzed. This is where the Process FMEA shows its simplicity. A work instruction is already a broken down linear list of step-by-step tasks needed to build the product. These steps are be copied into the FMEA worksheet in the 'Step' column.

Step #	Step Name	Failure Mode	Effects	Targets	Root Cause	S	O	D	RPN	Solution Actions	Current Methods
1	Clean the sheet metal with alcohol										
2	Place thermal tape on border										
3	Clean the bottom of the heat sink with alcohol										
4	Place heat sink on thermal tape and apply heavy pressure										
5	Clean the top of the heat sink with alcohol										
6	Place LED strip on top of the heat sink. Apply heavy pressure without placing pressure on LEDs										

Table 1 – FMEA Worksheet

The columns in Table 1 will be referenced throughout this section.

3.1 FMEA Steps

To perform a Process FMEA, simply follow these steps:

3.1.1 Define Failure Modes – *What can go wrong here?*

For every step previously defined, ask the question, "What can go wrong here?". The answer to this question needs to be focused directly on this step alone; assume everything else in the manufacturing process was completed correctly. An important note is that not all manufacturing process steps have a discernable failure mode. If no possible failure modes are present, skip to the next row of the worksheet.

Step: Clean the sheet metal with alcohol

Possible Failure: The sheet metal may not be fully cleaned

Possible Failure: The sheet metal could be bent during the cleaning process

These two possibilities are the failure modes of this step. They go under the "Failure Modes" column on the FMEA worksheet.

3.1.2 Define Effects – *What will happen then?*

The next step is to ask, “What will happen then?” for every failure mode listed. Look exclusively at only one failure mode at a time and focus on any direct results from the failure.

Failure Mode: The sheet metal was not fully cleaned

Direct Result: the thermal tape to be applied here could not fully adhere, leading to a breakdown of the thermal path which could lead to the PCB to overheat or to become dislodged entirely.

Failure Mode: The sheet metal could be bent during the cleaning process

Direct Result: an air gap between the heat sink and the sheet metal, resulting in a breakdown of the thermal path that could result in the PCB overheating.

Direct Result: a bend was severe enough to prevent installation at a later step. The result would be to scrap or rework that part.

These results are the effects on the system and go into the “Effects” column on the FMEA worksheet.

3.1.3 Describe Targets – *Who will suffer from the failure?*

The third step is to define the targets for each effect. This is done by asking the question, ”Who will suffer from the failure?”.

Effect: Tape not fully adhered, thermal path breakdown, PCB overheating

Who Suffers: End customer will suffer due to overheating generally taking a good deal of stress to cause issues and is likely not to be caught at manufacturing test. There is a good chance this defect would ship to the customer.

Effect: Air gap in thermal tape, thermal path breakdown, PCB overheating

Who Suffers: End customer will suffer due to overheating generally taking a good deal of stress to cause issues and is likely not to be caught at manufacturing test. There is a good chance this defect would ship to the customer.

Effect: Severe bend caused part to be unusable during install causing a scrap or rework of that part

Who Suffers: The company would suffer since this part would no longer physically fit into the system. The part would need to be reworked or scrapped, resulting in lower yields and higher operating costs.

These targets are captured the “Targets” column on the FMEA worksheet.

3.1.4 Find Root Causes – *Why will that happen?*

The fourth step is to define the root causes by asking, “Why will that happen?”.

Failure Mode: The sheet metal was not fully cleaned

Root Cause: A contaminant that is insoluble with alcohol

Root Cause: Operator error

Failure Mode: The sheet metal was bent during cleaning

Root Cause: Operator error

The root causes are captured in the “Root Causes” column on the FMEA worksheet.

3.1.5 Prioritize the Risks – *What is the likelihood?*

The fifth step is to prioritize the risks associated with the failure mode. Every failure is assigned three numeric ratings in a scale of 1 to 10:

- Severity (S): 1 (insignificant) to 10 (catastrophic)
- Likelihood of Occurrence (O): 1 (unlikely) to 10 (inevitable)
- Detectability (D): 1 (guaranteed to be detected) to 10 (undetectable).

These three numbers are multiplied to create the Risk Priority Number (RPN). Thus, S x O x D = RPN.

Effect: Tape not fully adhered, thermal path breakdown, PCB overheating

Severity: 6. Shortened PCB component lifespan.
Occurrence: 2. With proper training, this is unlikely to happen.
Detectability: 3. This can be easily detected in production.
RPN: 6 x 2 x 3 = 36

Effect: Air gap in thermal tape, thermal path breakdown, PCB overheating

Severity 6: PCB overheating would shorten the lifespan of components.
Occurrence 2: With proper training this is unlikely to happen.
Detectability 1: Very easily detected in production.
RPN: 6 x 2 x 1 = 12

Effect: Severe bend makes part unusable causing a scrap or rework

Severity: 3. Wasted time or resources. This will not directly affect the customer.
Occurrence: 2. With proper training, this is unlikely to happen.
Detectability: 1. Easily detected in production.
RPN: 3 x 2 x 1 = 6

The severity, occurrence, detectability, and RPN go on the FMEA worksheet.

3.1.6 Define Solution Actions – *How can this be prevented?*

The next step is to define solution actions. These are steps that can be taken to mitigate the chance of a particular failure occurring.

Failure Mode: The sheet metal may not be fully cleaned

Solution Action: Operator training on proper cleaning techniques.

Solution Action: Additional step in work instruction to check for contaminants.

Failure Mode: The sheet metal could be bent during the cleaning process.

Solution Action: Operator training on proper cleaning techniques.

Solution Action: A fixture may be needed to hold the sheet metal during the cleaning process.

Enter the solution actions into the “Solutions” column on the FMEA worksheet.

3.1.7 Describe Current Prevention and Detection Methods – *What is currently being done?*

The final step is to describe what is currently being done to prevent or detect the failure.

Failure Mode: The sheet metal may not be fully cleaned.

Current Methods: Operator training prior to working on the production line. Work instructions are displayed during production.

Failure Mode: The sheet metal could be bent during the cleaning process.

Current Methods: Operator training prior to working on the production line. Work instructions are displayed during production.

Enter the current methods into the “Current Methods” column on the FMEA worksheet.

3.1.8 Repeat

Repeat these steps for every row of the worksheet until it is complete. After the process is complete, the highest risk areas of the system can be identified by sorting on the RPN column. From this information, it is possible to identify where the most benefit will be realized from implementing mitigations or redundancy into the system.

Table 2 shows the completed FMEA worksheet for the first step of the manufacturing process. The above steps should be repeated for every row in the worksheet until it is completed.

Step #	Step Name	Failure Mode	Effects	Targets	Root Cause	S	O	D	RPN	Solution Actions	Current Methods
1	Clean the sheet metal with alcohol	May not be fully cleaned	thermal path breakdown. Possible PCB overheating. Reduced lifespan	End Customer	Contaminany insoluble with alcohol or operator error	6	2	3	36	Operator training on proper cleaning. Additional verification step for contaminants	Operator training prior to production. WI displayed during production
		Bent during cleaning	Air gap causes thermal path breakdown. Possible PCB overheating. Reduced lifespan	End Customer	Operator Error	6	2	1	12	Operator training on cleaning techniques.	Operator training prior to production. WI displayed during production
			Bend is severe enough to prevent installation.	Company	Operator Error	3	2	1	6	Fixture may be needed to hold metal during cleaning process	Operator training prior to production. WI displayed during production
2	Place thermal tape on border										

Table 2 – Completed Process FMEA Worksheet for Step 1

4. FMEA in a Software System

To perform a FMEA in a software system, the Functional FMEA model is used. The main difference from a Process FMEA is that for the Software FMEA there is no pre-defined, linear flow that can be copied from a work instruction. In the example in Section 3, the work instructions are the starting point for the FMEA steps. In a software system decisions are made at run-time to determine the appropriate path. This complicates the definition of the steps needed to begin the FMEA worksheet. Another fundamental difference between Process and Software FMEA is that, for software, it is not suggested to cover 100% of the design. Instead, focus efforts where failures are most likely or where the consequences are severe. These should be limited to a particular sub-system or even a new feature of a system.

In order to define steps, first break the system down into Function Blocks to create a Functional Block Diagram (FBD). A function block is a high level description of a piece of the software system. The level of detail that goes into the function blocks is a large decision and can be influenced by a variety of factors such as:

- New technologies or hardware present – The team may want to spend more time on areas where new technologies or hardware are present, so more detailed blocks should describe these areas.
- Overall confidence of the subsystem – Was there any re-used code in the system? These parts can be described with larger, less descriptive blocks, so more energy is focused where problems are more likely.
- Safety concerns – When safety is a concern, functional blocks should be more detailed so there is less chance for an oversight or assumption.

A natural approach might be to use the FBD from the design specification, but this can often lead to extensive and complicated analysis, so a higher-level FBD may need to be created. This can be accomplished by combining individual blocks of the FBD to make a more generalized version.

Once the generalized FBD is created for the system under analysis, it is likely that the diagram will not be linear and will not fit nicely into a worksheet. There may be a need to break up the FBD into smaller, more linear pieces so it can be easily referenced on a worksheet. Figure 1 provides an example of this.

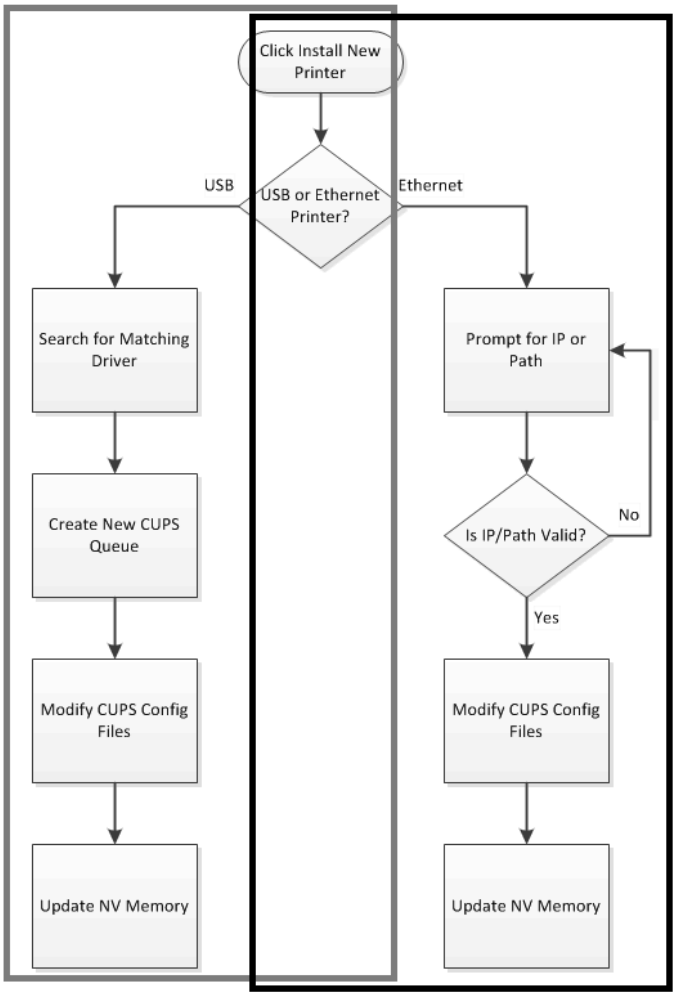

Figure 1 – FBD Splitting Example

Once the system is broken into smaller, more manageable FBDs, each can be placed into the FMEA worksheet. The FMEA worksheet will have a small section for each FBD so they can be easily referenced throughout and after the meeting. In the above example, there would be two sections for the FMEA worksheet. One would be titled “Installing USB Printer”, and the other would be titled “Installing Network Printer”. An example of this is below in figure 2.

Step #	Step Name	Failure Mode	Effects	Targets	Root Cause	S	O	D	RPN	Solution Actions	Current Methods
Installing USB Printer											
1											
2											
3											
4											
Installing Network Printer											
1											
2											
3											
4											

Figure 2 – Split FMEA Worksheet

Once the blocks are entered into the worksheet, go through the same FMEA steps (with one key difference) as the previous example, which were:

1. Define Failure Modes – *What can go wrong here?*
2. Define Effects – *What will happen then?*
3. Describe Targets – *Who will suffer from the failure?*
4. Find Root Causes – *Why will that happen?*
5. Prioritize the Risks – *What is the likelihood?*
6. Define Solution Actions – *How can this be prevented?*
7. Define Current Prevention and Detection Methods – *What is currently being done?*

The key difference between these steps in a Software FMEA versus the Process FMEA is that the root cause is often identical to the failure mode. There could be a root cause column in the FMEA worksheet, but it is not required when it is identical to the failure mode.

Note, while defining failure modes in a software system, it is important to not have ‘software bugs’ as a failure mode. Since any piece of code can contain bugs, this is a meaningless failure mode and should not be considered during this analysis. With that said, the FMEA meeting is a good time to see which blocks may be more likely to have bugs, yielding great candidates for an in-depth code reviews.

4.1 Software FMEA Example

As a Software FMEA example, consider a simplified purchasing algorithm for a website. The FBD is displayed in Figure 3 below.

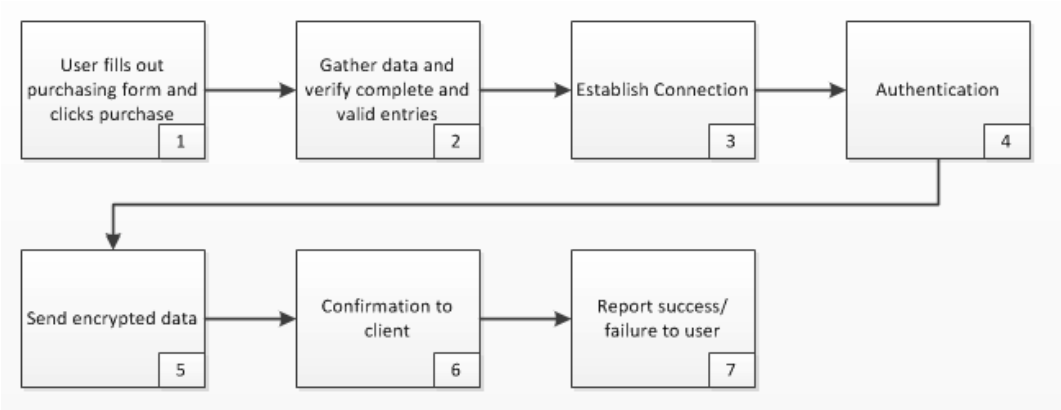

Figure 3 – Software FBD Example

A completed FMEA worksheet for the above FBD is located in Table 3 with highlights noted below.

Step #	Step Name	Failure Mode	Effects	Targets	Root Cause	S	O	D	RPN	Solution Actions	Current Methods
1	User fills out purchasing form and clicks purchase								0		
2	Gather data and verify complete and valid entries	SQL Injection appears as valid entry	Data theft or database corruption	Customers and Company	SQL injection not caught in verifying valid entries	10	3	2	60	Check for SQL injections in all fields	N/A
3	Establish connection with server								0		
4	Secure authentication with server								0		
5	Send encrypted data	User hits back/refresh causing data to be re-sent	multiple or incomplete orders being sent to the server	Customer	User hitting back/refresh prior to confirmation	7	5	1	35	Disallow multiple purchases within a specified timeframe from a specific user	Display a warning message to not hit back/refresh
6	Confirmation to client	Client model locked so it is not updated.	Incorrect status report to user	Customer and Company reputation	Semaphore placed on client model during update time	7	2	1	14	Have a queue to handle communication from server	N/A
7	Report success/failure to user								0		

Table 3 – Software FMEA Worksheet

Highlights from the above table:

- Two user-caused failure modes, and one possible development oversight.
- RPN column strongly reflects what would be most severe failures. Data theft at the top, and wrong confirmation at the bottom.
- Step 5 has a failure mode and root cause as being the same, which is acceptable in a Software FMEA.
- A few steps have no known failure modes. This is acceptable. Not all function blocks will have any discernable failures.
- Solution Action column can be added to software requirement specification to verify functionality.
- Test development is aided in steps 2 and 5 by performing or writing tests to attempt to recreate the failure modes.

4.2 Appropriate Time for a Software FMEA

The most benefit of a Software FMEA will be realized in early phases of design, ideally before any code has been written but after a majority of the requirements are defined. When performed early, the FMEA can reveal weak points of the system. Solutions can be designed in hardware, software, or both that can help avoid costly design changes in later project phases. Having the flexibility to determine the most cost-effective and most reliable solution sooner rather than later is the largest benefit of performing the FMEA.

Test development and planning can also be aided when performed early by knowing additional testing efforts up-front. These extra efforts may come from:

- Verification needed for all implemented solution actions
- Test cases can be created where the “Current Prevention/Detection Methods” column is empty.
- More testing efforts can be directed towards high RPN values.

While the most benefits of doing a Software FMEA are realized in the earliest design phases, many benefits can still be had in the later phases as well. The riskiest portions of the design will still be identified, and many solutions can still be developed. However, this typically comes at a cost of schedule or budget, and the flexibility to solve problems in hardware may be lost.

As an example, in an Anti-lock Braking System (ABS), there is a significant amount of hardware and software working together to keep the occupants of the vehicle safe. If a FMEA was performed on this system at any time, it may be realized that a single point of failure is in the sensor that detects when ABS needs to be applied. If this was caught early, there may be time to add in a redundant sensor and design a fail-safe of mechanical braking if both sensors fail. If caught too late, the redundant sensor may not be

an option due to schedule constraints, and the only option would be to revert to mechanical braking if the sensor fails. Hence, the risk of this failure occurring would be reduced if caught early.

4.3 Mechanics of the Software FMEA Meeting

Since the FBDs to be used for the analysis are not created by the entire team attending the FMEA meeting, it is likely that there will be some modifications during the meeting. For this reason, it is recommended to display the FBDs in a way where they can be quickly re-arranged during the meeting if it is desired. A group of blocks may be grouped together, or one block may be split into several, depending on the level of detail desired. A software tool, such as Microsoft Visio can be used, or simple post-it notes can be placed on a wall or whiteboard to display the FBDs during the meeting.

The meeting discussion should be maintained at a high level. When a group of engineers get into a discussion about issues and continue asking questions that begin with “what if”, the conversation often drills down to a very low level very quickly. Although these discussions are useful, and they should be had, during the meeting they should simply be noted and then continue with the meeting at a high level. Keep in mind that this is not a code or design review.

5. Conclusion

FMEA is a widely used and accepted method of reliability engineering. Its purpose is to identify possible failures, evaluate their effect on the system, and propose solutions to mitigate these effects. FMEA is most commonly used on the process and hardware levels, but is becoming more commonplace in the software industry. Some benefits of using FMEA on a software system include:

- Early identification of single failure points and system interface problems
- Getting a better idea for planning/scheduling
- Assisting in early test planning
- Catching issues early enough to be solved in hardware of software
- Single worksheet summary of the analysis

There are also fringe benefits to ‘what if?’ thinking. Having a dedicated time to get a group of engineers together to think about ways the system could break down leads to valuable test cases, and eventually, more reliable software.

Some common mistakes made during Software FMEAs include:

- Assuming that all failure modes are caused by hardware.
- Attempting to cover 100% of the design instead of focusing on where the design is most likely to cause serious failure.
- Neglecting to follow through with solution actions discovered during the process.
- Allowing the meeting to get into a low level (or code level) conversation.
- Not having appropriate domain experts attend the meeting.

A FMEA is a cost-effective way to guide you to a more reliable design by highlighting the most valuable places to insert fail-safes, redundancies, or other elements that increase the system’s overall reliability.

References

Softrel, LLC. “SFMEAs and SFTAs” FMEA. http://www.softrel.com/fmea.htm (accessed 7/12/2013).

Michael Barr “Building Reliable and Secure Embedded Systems” Embedded. http://www.embedded.com/electronics-blogs/barr-code/4238429/Building-reliable-and-secure-embedded-systems (accessed 6/17/2013).

Mike Silverman and George de La Fuente. “Software Design for Reliability” Ops A La Carte LLC. http://www.opsalacarte.com/pdfs/Tech_Papers/Software_Design_for_Reliability_-_Paper.pdf (accessed 7/10/2013).

Turabian, Kate. 2007. *A Manual for Writers of Research Papers, Theses, and Dissertations: Chicago Style for Students and Researchers*. Chicago: The University of Chicago Press.