

Software FMEA Toolkit Tutorial

SoftRel, LLC

www.softrel.com

amneufelder@softrel.com

© Softrel, LLC 2016

This presentation may not be copied in part or in whole without
written permission from support@softrel.com

Help

Every worksheet has at least one online help file link to guide you through the toolkit.

Additional resources

- Your toolkit has online help for every worksheet
- Each worksheet has “call outs” to guide you
- The toolkit has been designed to work with the separately sold book ["Effective Application of Software Failure Modes Effects Analysis"](#)

Installation

- The toolkit installation is illustrated in the below video

<http://www.softrel.com/images/installation.mp4>

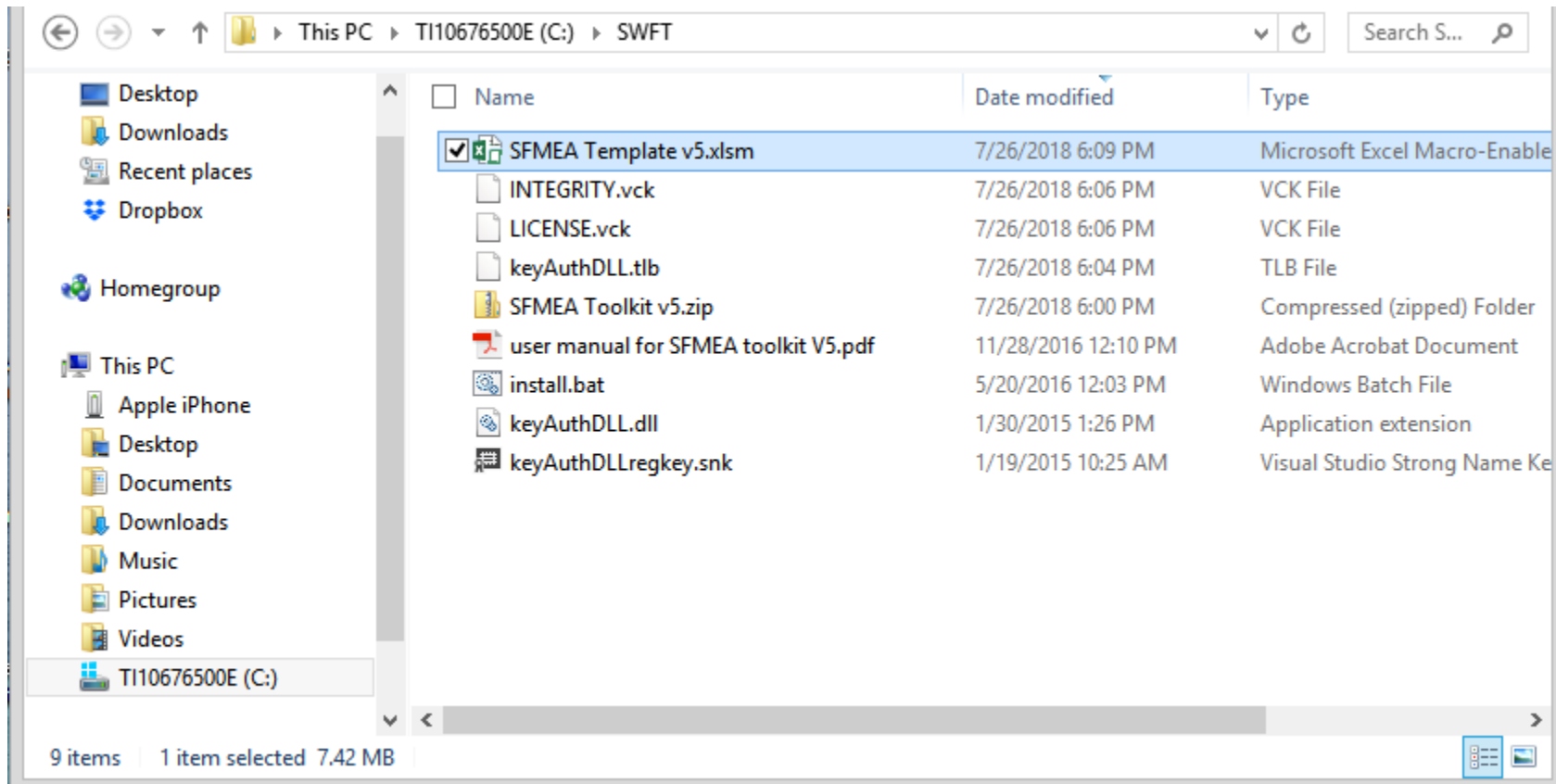
Step 1. Get started

The toolkit is a macro enabled spreadsheet

You must have macros enabled in Microsoft Excel© to use the toolkit

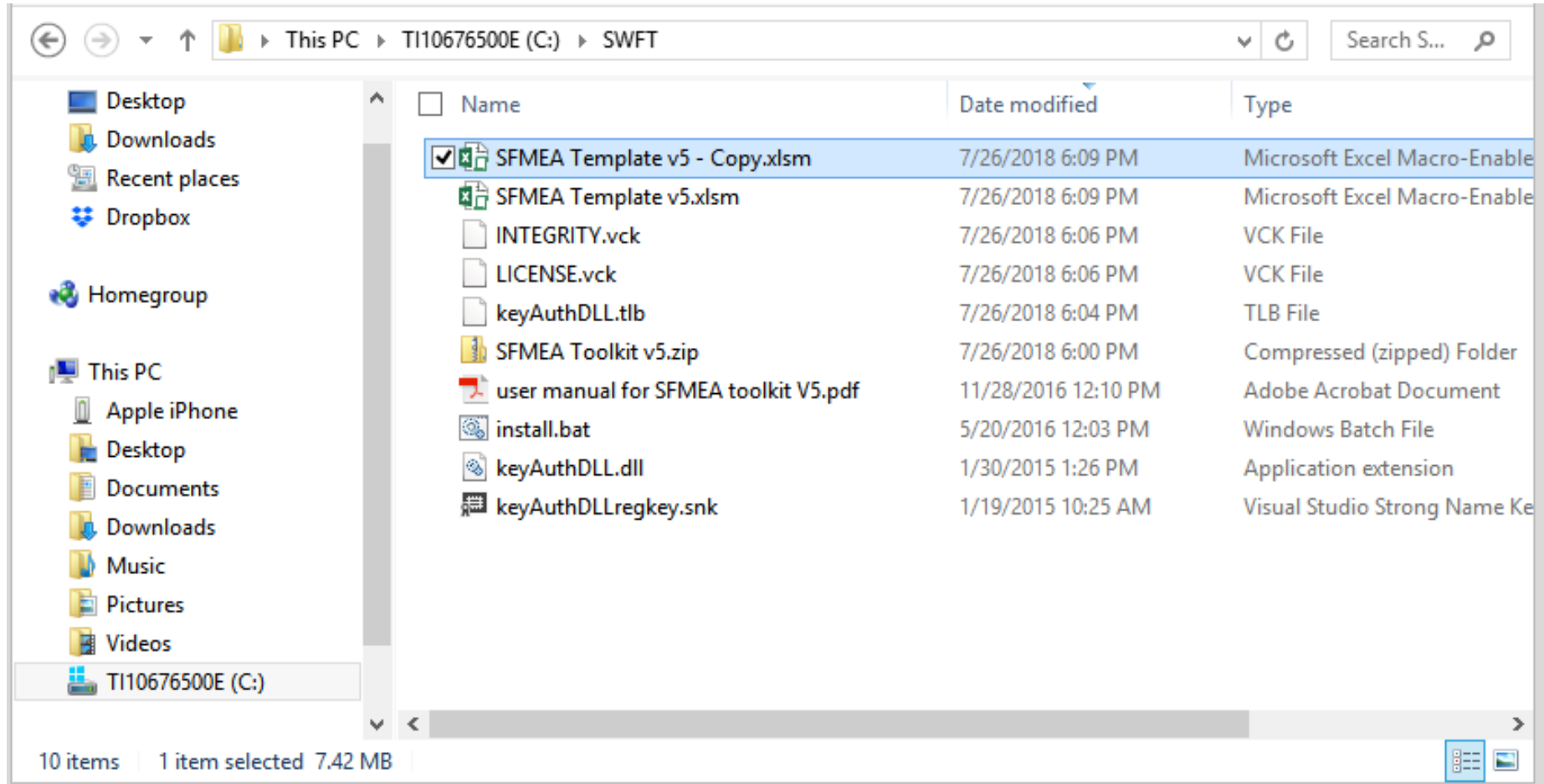
Opening the toolkit

- Prior to launching the software reliability toolkit you must
 - Have a recent version of Microsoft Excel
 - Make sure that the zip file is unzipped to c:/SWFT folder (note the files that should be extracted in the below figure)
 - Enable macros in Microsoft Excel
 - Activate the license
- Then launch the toolkit by simply selecting the macro enabled file and opening it with Microsoft Excel

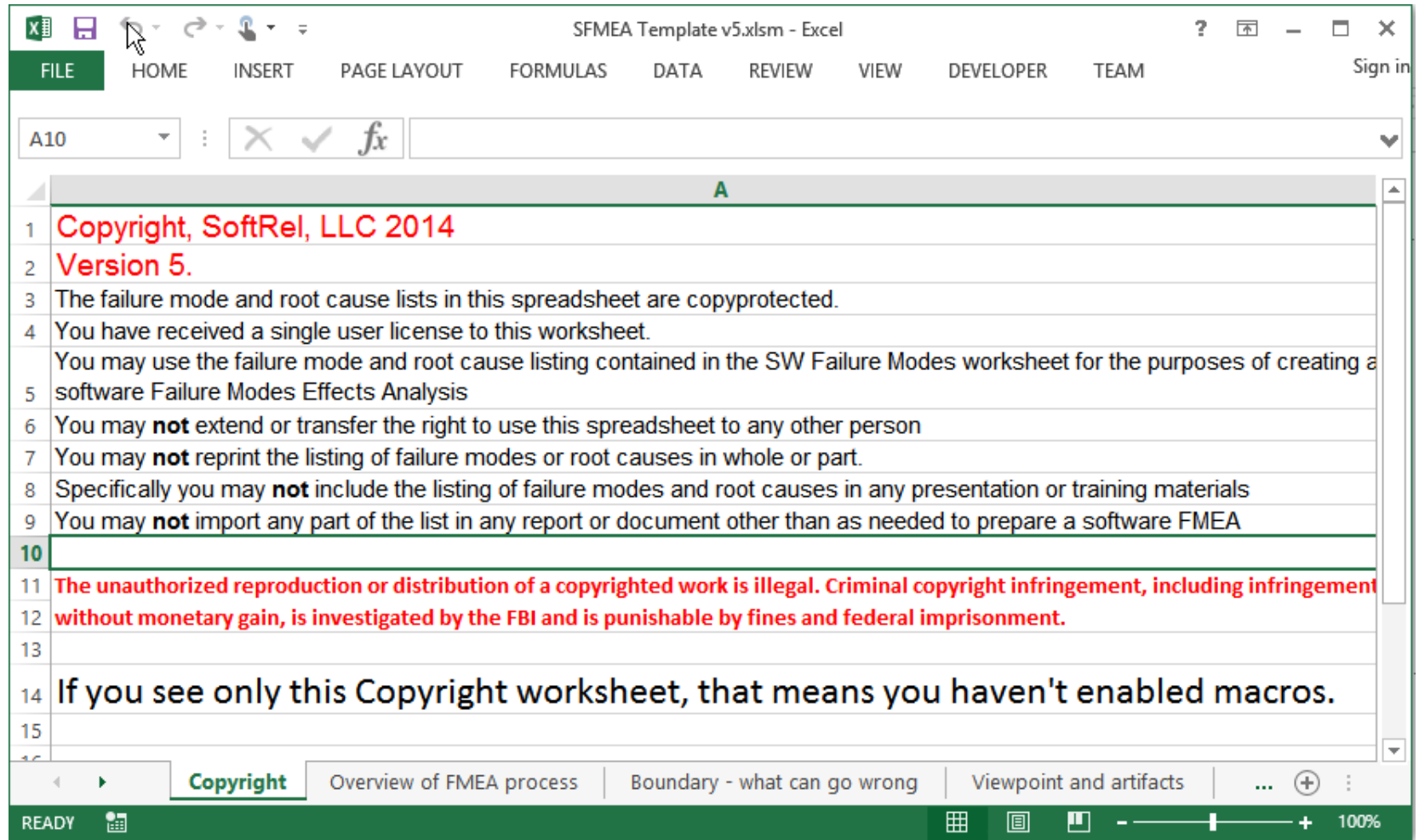


Copying the toolkit

- The “Save As” is not an allowed feature for the toolkit so to create multiple SFMEAs from template use the File Manager to copy and paste.
- You can make as many templates as you like as long as they remain in the SWFT folder.

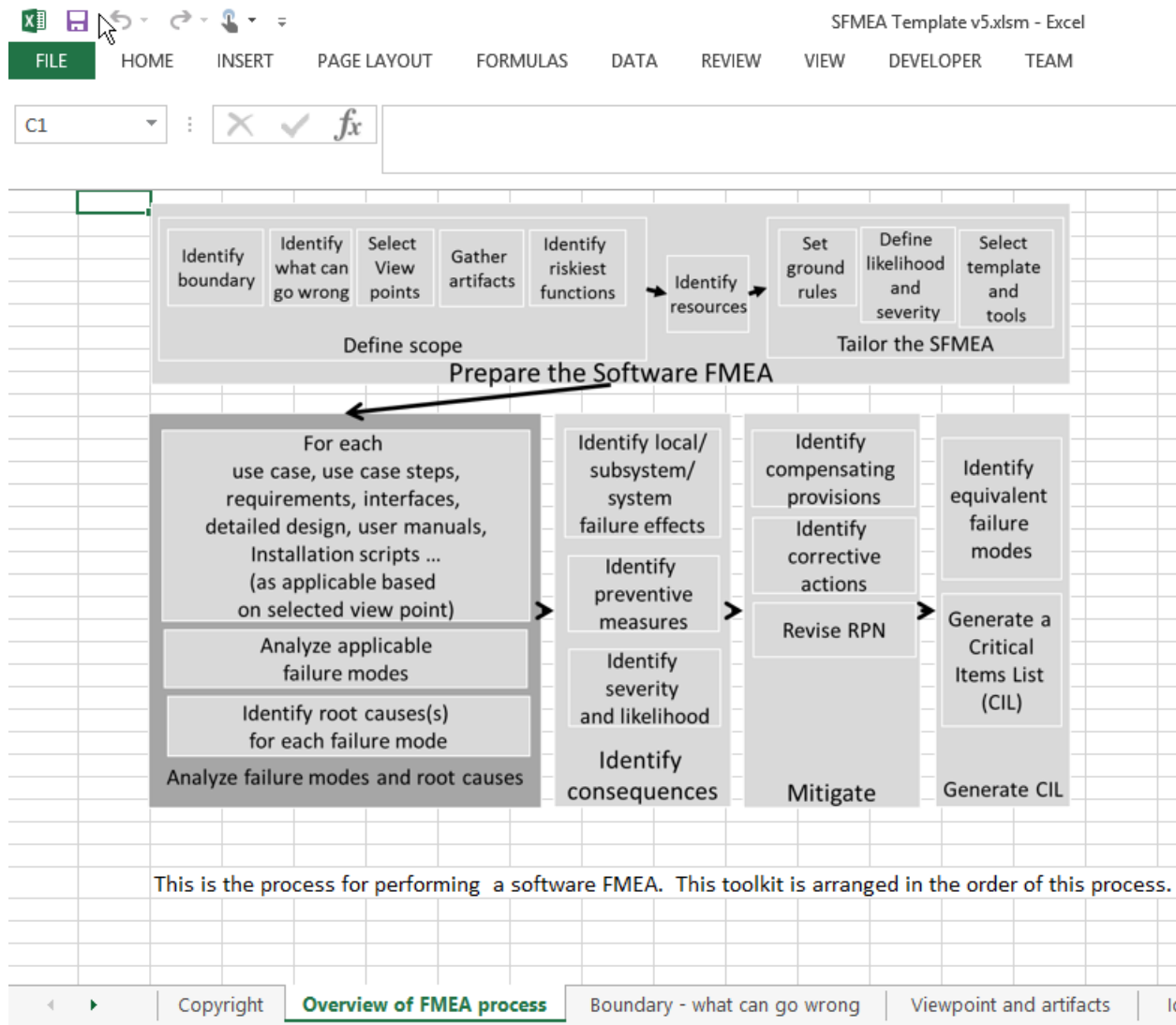


Copyright



- The toolkit is a single user/computer license.
- Read the Copyright notice
- If you see only the Copyright worksheet when you open the toolkit then you haven't enabled the macros.

Overview of the SFMEA



- The Overview page summarizes the rest of the toolkit
- Each step of the SFMEA is presented in order from left to right in each of the toolkit worksheets

Step 2. Prepare the SFMEA

Define the scope and resources and tailor the SFMEA template

These videos illustrate the SFMEA preparation worksheets

<http://www.softrel.com/images/preparesfmea1.mp4>

<http://www.softrel.com/images/preparesfmea2.mp4>

Prepare the SFMEA

- 2.1 Identify boundary
- 2.2 Identify what can go wrong
- 2.3 Select viewpoints
- 2.4 Gather artifacts
- 2.5 Identify riskiest functions
- 2.6 Identify resources
- 2.7 Set ground rules
- 2.8 Define likelihood and severity
- 2.9 Select template and tools

Identify boundary

The screenshot shows an Excel spreadsheet with the following content:

	A	B	C	D	E	F	G	H	I	J
1	Step 1. Identify the boundary of the system									
2	The SFMEA can be conducted on a component, element or system of systems									
3	Identify the system and architectural boundaries of the SFMEA here.									
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										

The first step is to decide the boundary of the system under analysis.
Will the boundary be a component of a system, element of a system, system of systems?
Whatever is outside of the boundary identified here will not be analyzed.

Identify what can go wrong

SFMEA Template v5.xlsm - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER TEAM

E16

	A	B	C	D	E	F	G	H	I	J	K
16	Step 2. Identify what can go wrong										
17	1. The software engineer won't write code that's not specified. So, if something important is not in writing, it can b										
18	2. The software test engineers won't test code that's not specified. So, if something important is not in writing, it ca										
19	3. It's possible that the software engineer won't even write code that is specified										
20	4. It's possible that the software engineer will write the wrong code to a correct specification										
21	5. The software engineer may guess if not given enough information										
22	6. It cannot be assumed that all failure modes will be identified via "thorough testing". Unless a specific failure mo										
23	7. Identify the top level failure events										
24	a. If a hardware FMEA has been performed, locate it. The element and system level effects for the hardware fa										
25	b. Brainstorm what can go wrong at the system, element and component level with regards to the software.										
26	c. Research past element and system level effects as well as failure modes and root causes on similar systems.										
27	d. In addition or alternatively, a system fault tree analysis is quite useful for identifying element and system le										
28											
29	An effective SFMEA will ultimately map the software failure modes to many of the below identified events. If the SFM either the SFMEA is a deep SFMEA with regards to a small number of critical functions or the SFMEA hasn't been prepa										
30	Past failure events on similar systems caused by software										
31											
32											

READY

Copyright Overview of FMEA process **Boundary - what can go wrong** Viewpoint and artifacts

Scroll down from the boundary section and the “Identify what can go wrong” section is shown

Some things that can go wrong...

The SFMEA analyst should remember the below when analyzing the failure modes and effects

1. The software engineer won't write code that's not specified. So, if something important is not in writing, it can be assumed the code won't be written to address it.
2. The software test engineers won't test code that's not specified. So, if something important is not in writing, it can be assumed the test plan won't test it.
3. It's possible that the software engineer won't even write code that is specified
4. It's possible that the software engineer will write the wrong code to a correct specification
5. The software engineer may guess if not given enough information
6. It cannot be assumed that all failure modes will be identified via "thorough testing". Unless a specific failure mode in a specific use case or function is explicitly in the test plan, it should be assumed it won't get tested.

Identify specifically what can go wrong with the system under analysis

Identify the top level failure events

- a. If a hardware FMEA has been performed, locate it. The element and system level effects for the hardware failures are often system level effects for software failures as well.
- b. Brainstorm what can go wrong at the system, element and component level with regards to the software.
- c. Research past element and system level effects as well as failure modes and root causes on similar systems.
- d. In addition or alternatively, a system fault tree analysis is quite useful for identifying element and system level effects and failure modes and root causes

An effective SFMEA will ultimately map the software failure modes to many of the below identified events. If the SFMEA only maps to a few of these events then either the SFMEA is a deep SFMEA with regards to a small number of critical functions or the SFMEA hasn't been prepared as effectively as it can be.

Use the template to define all of the failure events from a through d

Select viewpoints

SFMEA Template v5.xlsm - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER TEAM

A3

	A	B	C	D	E	F	G
1	Step 3. Select Viewpoints. These are the viewpoints. Identify which ones are relevant and appropriate for the current phase.						
2		FMEA	When this viewpoint is relevant	The earliest that this SFMEA can be performed...	Artifacts required		
3		Functional	Any new system or any time there is a new or updated set of requirements or use cases.	End of software requirements phase	Use Cases, Software Requirements Specification (SRS) or Systems Requirements Specification (SyRS)		
			Anytime there is complex hardware and software				

Copyright | Overview of FMEA process | Boundary - what can go wrong | **Viewpoint and artifacts**

Select destination and press ENTER or choose Paste

Select the “Viewpoint and artifacts” tab. Review each of the viewpoints, when it is relevant, the earliest that that viewpoint can be analyzed and the artifacts required for that viewpoint. Select the viewpoint(s) that are most relevant, applicable to the current phase of development and for which you have the artifacts to support. Generally 1 or 2 viewpoints are selected.

Gather artifacts

- Selecting the viewpoint and gathering the artifacts are performed iteratively until it is decided which viewpoint is most relevant and applicable.
- Tab down and review the recommended and required artifacts for each viewpoint. Select “Y” or “N” for each artifact. The toolkit will advise you which viewpoint you have sufficient information to analyze.

11	Step 4. Gather artifacts - verify selected viewpoint is possible given artifacts							
12	1. Identify which artifacts will be available for the SFMEA. Select Y for Yes or N for N in the second column.							
13	2. The viewpoints which are feasible given the available artifacts are shown in row 6.							
14	3. For each viewpoint the artifacts that are required, highly recommended and optional artifacts are listed.							
15	Key	Y - Required	H - Highly recommended	O - Optional				
16			Functional					
17	Artifacts/Viewpoints	Available?	System of system	Use cases within an element	Use Case	SRS	Interface	
18	These viewpoints are applicable		Not enough information	Not enough information	Not enough information	Yes, the artifacts are sufficient.	Not enough information	No in
19	Software Requirements Spec	Y	H	H	H	Y	Y	
20	System requirements spec	N	Y	H	H	H	H	
21	System Architecture Design	N	Y	H	H	H	H	
	Interface Control Spec		Y	H	H	O	Y	
		Copyright	Overview of FMEA process	Boundary - what can go wrong	Viewpoint and artifacts			

Identify riskiest functions

1	Step 5. Identify riskiest functions											
2	The use case, interface, detailed, maintenance, vulnerability and usability FMEAs require pruning of scope for an effective SFMEA											
3	For the selected viewpoint, identify the risk level of each use case, interface or function name and decide whether it will be in scope based on the final risk level.											
4		Use Case					Interface					
	Applicable viewpoints for this use case, interface or function		Mission				Mission		Mission			
			Safety rating	impact rating	Development risk	In scope?	Interface from	Interface To	Safety rating	impact rating	Development risk	In scope?
5		Use Case										
6	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
7	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
8	Functional system of sy:		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
9	Functional Use Case		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
10	Functional SRS		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
11	Interface		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
12	Detailed		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
13	Maintenance		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
14	Usability		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
15	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
16	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
17	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
18	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
19	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
20	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N

Select the “Identify riskiest functions” tab. Go to the first column and select which viewpoint you plan to analyze. You may have more than one viewpoint so each row can be used to identify the riskiest function by viewpoint.

If the use case is selected, for example, type the names of each use case in the second column.

Identify riskiest functions

	A	B	C	D	E	F	G	H	I	J	K	L
1	Step 5. Identify riskiest functions											
2	The use case, interface, detailed, maintenance, vulnerability and usability FMEAs require pruning of scope for an effective SFMEA											
3	For the selected viewpoint, identify the risk level of each use case, interface or function name and decide whether it will be in scope based on the final risk level.											
4		Use Case					Interface					
	Applicable viewpoints for this use case, interface or function	Use Case	Safety rating	Mission impact rating	Development risk	In scope?	Interface from	Interface To	Safety rating	Mission impact rating	Development risk	In scope?
5												
6	Unsure	Name of use cas here	Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
7	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
8	Unsure		Low	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
9	Unsure		Medium	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
10	Unsure		High	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
11	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
12	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
13	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
14	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
15	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
16	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
17	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
18	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
19	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N
20	Unsure		Unsure	Unsure	Unsure	N			Unsure	Unsure	Unsure	N

... Boundary - what can go wrong Viewpoint and artifacts **Identify riskiest functions** Resources and groundrules Severity Likeli ...

For each use case select the appropriate safety and mission risk. The preliminary hazards analysis may be used for this assessment. The development risk is an indication of how stable/complex the particular use case is.

Identify riskiest functions

1	Step 5. Identify riskiest functions					
2	The use case, interface, detailed, maintenance, vulnerability and usability FMEAs require pruning of s					
3	For the selected viewpoint, identify the risk level of each use case, interface or function name and de					
4	Use Case					
	Applicable viewpoints for this use case, interface or function		Safety rating	Mission impact rating	Development risk	In scope?
5	Use Case					Interface from
6	Unsure	Name of use cas here	High	Unsure	Unsure	N
7	Unsure		Unsure	Unsure	Unsure	Y
8	Unsure		Unsure	Unsure	Unsure	N
9	Unsure		Unsure	Unsure	Unsure	N
10	Unsure		Unsure	Unsure	Unsure	N
11	Unsure		Unsure	Unsure	Unsure	N
12	Unsure		Unsure	Unsure	Unsure	N
13	Unsure		Unsure	Unsure	Unsure	N
14	Unsure		Unsure	Unsure	Unsure	N
15	Unsure		Unsure	Unsure	Unsure	N
16	Unsure		Unsure	Unsure	Unsure	N
17	Unsure		Unsure	Unsure	Unsure	N
18	Unsure		Unsure	Unsure	Unsure	N
19	Unsure		Unsure	Unsure	Unsure	N
20	Unsure		Unsure	Unsure	Unsure	N

Boundary - what can go wrong
Viewpoint and artifacts
Identify riskiest functions

Once all use cases are assessed for risk, select which ones will be in scope for the analysis. This decision is a group effort based on time and resources available.

The other viewpoints work similarly to the use case viewpoint. In the case of the interface viewpoint, for example, the most risky interfaces are assessed.

Identify resources

		Name	Contact info	Estimated time f person
1	Step 6. Identify resources			
2	SFMEA Facilitator			
3	Software or Firmware Engineer			
4	Software Architect			
5	Software Requirements Engineer			
6	Software Manager			
7	Software Testing			
8	Domain Experts			
9	Safety Engineers			
10	Systems Engineers			

◀ ▶ ...
Viewpoint and artifacts
Identify riskiest functions
Resources and groundrules

- Select the “Resources and groundrules” tab
- Identify who will be performing the SFMEA.
- The SFMEA is not a “one person” analysis.
- Ensure that there are appropriate subject matter experts for the selected viewpoints.
 - All viewpoints require an analyst who understand software and software failure modes.
 - The detailed, maintenance and vulnerability viewpoints require someone who is intimately familiar with the software under analysis.

Identify the ground rules

	A	B	C	D
11	Step 7. Identify groundrules			
12		Issue	Extent the failure mode is propagated	Our decision
13		Human error	Decide whether or not to include human errors in the Functional SFMEAs. The Usability SFMEA focuses on the human error. However, it's possible to include the human aspect in the Functional SFMEA also.	
		Chain of interfaces	If an interface FMEA is selected, how many interface	
<div> ◀ ▶ ... Viewpoint and artifacts Identify riskiest functions Resources and groundrules </div> <div> READY 📅 </div>				

- Scroll down to the “Identify groundrules” area
- Review the ground rules and make decisions for this SFMEA with regards to consideration of
 - human error (with human error be included?)
 - interface chains (applicable only for interface viewpoint)
 - Seasonality (particular times of day, week, month, year in which the software is stressed differently than others)
 - Network availability (maximum, typical?)
 - Speed/throughput (maximum, normal?)
- These groundrules will be adhered to in the forthcoming analysis

Identify severity and likelihood

	A	B	C	D	E	F	G
1	Step 8. Identify the severity and likelihood ratings. The below are only examples. Use the severity and likelihood ratings defined by you						
2		Severity	Likelihood				
3	1	Catastrophic	Likely				
4	2	Critical	Reasonably Probable				
5	3	Marginal	Possible				
6	4	Minor	Remote				
7	5		Extremely unlikely				
8	Define the Failure Definition Scoring Criteria						
9	List specific examples of each severity level. The examples should cover the entire range of possible outcomes or hazards.						
10	The examples should be as specific as possible for the software and product under analysis. Avoid general hazards and focus on the spe						
	Severity	Project specific examples and criteria					
11							
12	1						
13	2						
14	3						
15	4						
16							
17	Define thresholds for mitigation. Adjust the color coding below to meet the needs of the project. This below is only an example.						
18	Likely	Minor	Marginal	Critical	Catastrophic		
19	Reasonably Probable						
20	Possible	High	High	Extreme	Extreme		
21	Remote	Moderate	High	High	Extreme		

- Identifying the severity and likelihood ratings (as military or other standards) is the easy part
- Identifying *concrete and specific* definitions of each is the difficult part
- The FDSC (Failure Definition Scoring Criteria) is a great way to assign **program specific events** to the severity levels. Identifying these up front can minimize time spent later in the analysis.

Step 3. Analyze failure modes and root causes

The toolkit comes with hundreds of software failure modes and root causes

These videos illustrate the failure modes analysis worksheets

<http://www.softrel.com/images/worksheets3.mp4>

<http://www.softrel.com/images/failuremodes4.mp4>

<http://www.softrel.com/images/failuremodes5.mp4>

Analyze failure modes and root causes

- These are the viewpoints which each have unique failure modes and root causes as well as unique templates to support the viewpoint under analysis
 - Functional
 - System of system
 - Use case
 - Use case steps
 - Functional SRS
 - Interface
 - Detailed/Vulnerability
 - Maintenance
 - Usability
 - Serviceability
 - Vulnerability
 - Production
- For each viewpoint the toolkit provides
 - A worksheet to add in project specific root causes in addition to the pre-populated root causes
 - A SFMEA worksheet
- Use only the worksheets for the viewpoints selected in the preparation phase

Functional SOS or overall use case viewpoint

	A	B	C	D	E	F	G
4	Subsystem or element	Component	Use cases	Common functions	Common messages	Potential failure mode	Potential root cause
5	Template for failure modes that span across use cases						
6	List all subsystems or elements covered by the use cases	List component names covered by use cases here	List all use cases here			Faulty functionality overall	Listing of faulty functionality span multiple use cases
7	Expand each failure mode (by copy and paste) as illustrated here with faulty functionality. Do not expand root causes that are not applicable. Insert rows for each applicable					Faulty functionality overall	Listing of faulty functionality span multiple use cases
8						Faulty processing overall	Listing of faulty processing span multiple use cases
9						Faulty timing overall	Listing of faulty timing root causes span multiple use cases
10						Faulty sequencing overall	Listing of faulty sequence root causes span multiple use cases
11						Faulty error handling overall	Listing of faulty error handling causes
12						Faulty state management overall	
13						Other failure modes spanning across use cases	
14						Faulty functionality overall	
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							
92							
93							
94							
95							
96							
97							
98							
99							
100							

Go to the “Overall use cases SFMEA” worksheet. The templates are prepopulated for failure modes that span across use cases or SOS. You can copy and paste the failure mode rows as well as delete them.

There is a pulldown menu for the failure modes and the root causes. You can also not use the template at all and insert a free form row.

The “overall use cases SFMEA” has 4 templates for 1) failure modes that span all use cases 2) common functions 3) common messages and 4) user related failure modes

Functional System of System (overall use cases)

B		C	
→		These failure mode root causes are in scope for this SFMEA	
2	Functional Viewpoint		
3	Listing of failure modes that span multiple use cases or multiple elements		
4	Listing of faulty functionality span multiple use cases	Possibly	<i>Applicability</i>
8	Enter another functionality root cause here	Possibly	
9	Enter another functionality root cause here	Possibly	
10	Listing of faulty processing span multiple use cases	Possibly	
12	Enter another processing root cause here	Possibly	
13	Enter another processing root cause here	Possibly	
14	Listing of faulty timing root causes span multiple use cases	Possibly	
16	Enter another timing related root cause here	Possibly	
17	Enter another timing related root cause here	Possibly	
18	Listing of faulty sequence root causes span multiple use cases	Possibly	
20	Enter another sequencing related root cause here	Possibly	
21	Enter another sequencing related root cause here	Possibly	
22	Enter another sequencing related root cause here	Possibly	
23	Listing of faulty error handling causes that span across use cases	Possibly	
32	Enter another error handling related root cause here	Possibly	
33	Enter another error handling related root cause here	Possibly	
34	Listing of faulty state management causes that span across use cases	Possibly	
Resources and groundrules		Severity Likelihood	FM SOS or overall use cases
			Overall use cases SFMEA

Go to the “FM SOS or overall use cases” worksheet. Add any known project specific root causes that apply to the entire system of system or all use cases. They will be added to the pre-populated pull down menus for this viewpoint.

Functional SOS or overall use case viewpoint

	A	B	C	D	E	F	G	
4	Subsystem or element	Component	Use cases	Common functions	Common messages	Potential failure mode	Potential root cause	Lo
5	Template for failure modes that span across use cases							
6	List all subsystems or elements covered by the use cases	List component names covered by use cases here	List all use cases here			Faulty sequencing overall	Listing of faulty functionality span multiple use cases	
7	Expand each failure mode (by copy and paste) as illustrated here with faulty functionality. Delete failure modes that don't apply to the u						Listing of faulty functionality span multiple	
8						Faulty functionality overall	The use cases conflict with each other	
9						Faulty functionality overall	Missing functions altogether	
10						Faulty functionality overall	Features that should be automated are not	
11							Enter another functionality root cause here	
12						Faulty processing overall	not (in any case use)	
13						Faulty timing overall	Listing of faulty processing span multiple use cases	
14						Faulty sequencing overall	Listing of faulty timing root causes span multiple use cases	
15							Listing of faulty sequence root causes span multiple use cases	
16							Listing of faulty error handling causes	

READY

Resources and groundrules Severity Likelihood FM SOS or overall use cases Overall use cases SFMEA FM Use case Us ... (+) 100%

The root causes for that failure mode are selected from a pull down menu. Any specific root causes that you entered will be displayed.

The root causes are synchronized with a particular failure mode.

So, copy and paste the rows for each failure mode to keep the failure modes and root causes in sync.

Worksheet capabilities

- Modify the text in the template headings
- Add or delete columns
- Add rows that don't have pull-down menus for free form analysis
- Tailor the list of root causes in the pull down menus (by modifying the FM (failure modes) worksheet associated with the viewpoint
- If you decide to create a company specific template, you can use that to copy and paste new worksheets.
- You can create as many worksheets as you want as long as they are saved in the c:/SWFT folder
- The worksheet templates can be emailed to other users who have a SFMEA license
- For users who do not have a license, you can print your worksheets and email.
- The “save as” feature is disabled. However, you can copy and paste the templates using the file explorer copy and paste commands.

Other viewpoint worksheets

- The other viewpoints work similarly to the Functional Overall Use Case SFMEA
- The failure mode worksheets and corresponding SFMEA worksheets are shown below for each viewpoint

Viewpoint	Failure mode worksheet	SFMEA worksheet
Functional - System of System Overall Use Cases	FM SOS or Overall Use Cases	Overall Use Cases
Functional – Use Case	FM Use Case	Use Case SFMEA
Functional – Use Case Steps	FM Use Case Steps	Use Case Steps SFMEA
Functional SRS – Software Requirements Specification	FM SRS	SRS SFMEA
Interface Design	FM Interface	Interface SFMEA
Detailed Design	FM DetVuln	Detailed SFMEA
Vulnerability		
Maintenance		Maintenance SFMEA
Usability	FM Usability	Usability SFMEA
Serviceability (installation scripts, update)	FM Serviceability	Serviceability SFMEA
Production (Process)	FM Production	Production

Step 4. Identify Consequences

Identify the effects on the software and the system and any preventive measures

This video illustrates the consequences and mitigations

<http://www.softrel.com/images/consequencesmitigation7.mp4>

Identify Consequences

<div> <div>Insert new rows for each of the root causes for the potential failure mode. Not all root causes are relevant for every common function and not all failure modes are relevant for every common function. Delete the failure modes that aren't applicable. Insert rows for each applicable root cause.</div> <div>those that don't apply after each has been analyzed</div> </div>										
4	Common messages	Potential failure mode	Potential root cause	Local Effect	Subsystem effect	System effect	Preventive measures	Severity	Likelihood	RPN
5										
6		Faulty sequencing overall	Listing of faulty functionality span multiple use cases							0
7	e with faulty functionality. Delete failure modes that don't apply to the use case. Don't expand root causes that aren't applicable.									
8		Faulty functionality overall	The use cases conflict with each other							
9		Faulty functionality overall	Missing functions altogether							
10		Faulty functionality overall	Features that should be automated are not (in any case use)							
11										
12		Faulty processing overall	Listing of faulty processing span multiple use cases							0
13			Listing of faulty timing root causes							
<div> <div>FM SOS or overall use cases</div> <div>Overall use cases SFMEA</div> <div>FM Use case</div> <div>Use case SFMEA</div> <div>FM use case steps</div> <div>Use case steps SFMEA</div> <div>...</div> </div>										
<div> <div>Cell F4 commented by AMN</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>100%</div> </div>										

Once the failure modes and root causes are analyzed, scroll to the right to analyze the effects on the software (local) and system. If there are any measures to prevent the failure mode, identify.

Tip: It's usually best to analyze all of the effects and preventive measures first and then analyze the severity and likelihood in one last pass.

The Risk Probability Number (RPN) is automatically calculated.

Step 5. Identify Mitigations

Identify corrective actions, compensating provisions and revised RPN

This video illustrates the consequences and mitigations

<http://www.softrel.com/images/consequencesmitigation7.mp4>

Identify Mitigations

	Local Effect	Subsystem effect	System effect	Preventive measures	Severity	Likelihood	RPN	Corrective actions	Compensating provisions	Severity	Likelihood	RPN	Test procedure?	Action item?	Fault insertion?
4															
5															
6							0					0			
7	aren't applicable.														
8															
9															
10															
11															
12							0					0			
13															

◀ ▶ ...
FM SOS or overall use cases
Overall use cases SFMEA
FM Use case
Use case SFMEA
FM use case steps
Use case steps SFMEA

READY
📅
🔍
📄
🖨
⌵

Once the consequences are identified, scroll to the right to analyze the corrective actions. If there are compensating provisions then identify those. Re-assess the severity and likelihood once the corrective actions are approved.

Corrective actions include but aren't limited to changing the requirements, design, code, test plan, user manual, installation guide, use case, etc.