Software FMEA Toolkit Tutorial

Ann Marie Neufelder

SoftRel, LLC

www.softrel.com

amneufelder@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

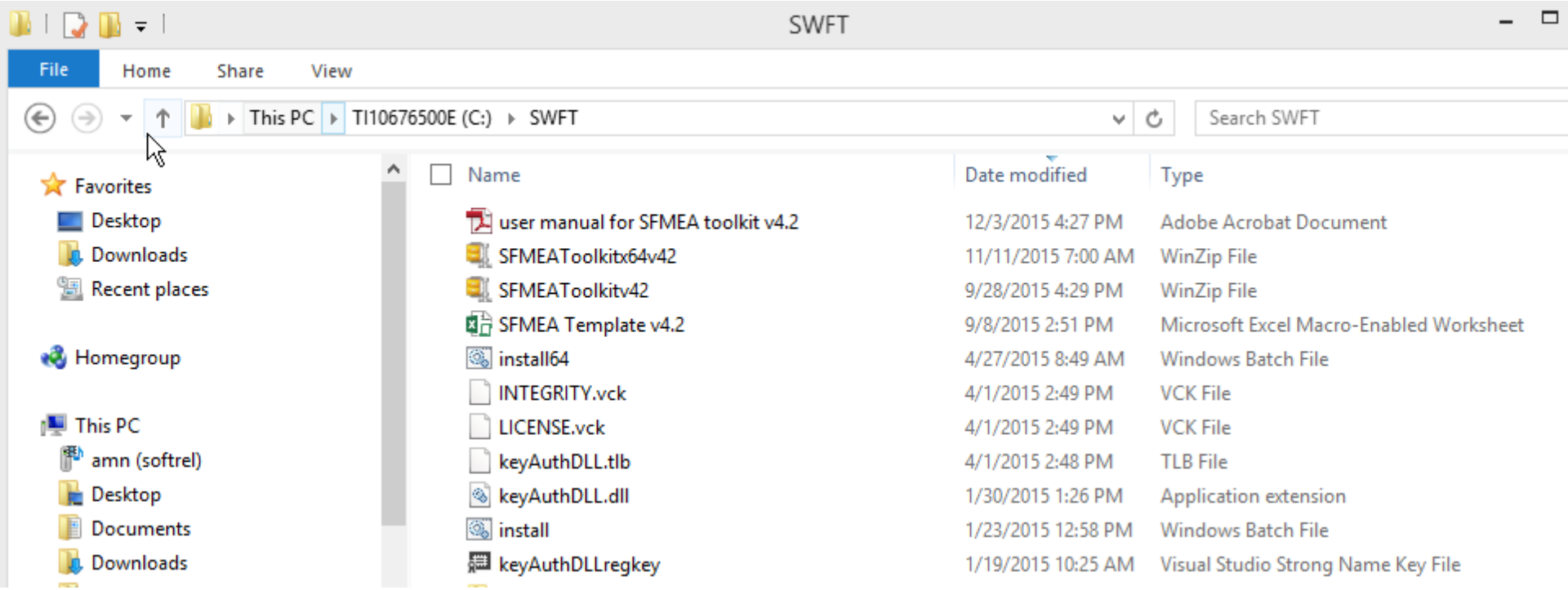
"Effective Application of Software Failure Modes Effects Analysis"

Step 1. Get started

The toolkit is a macro enabled spreadsheet

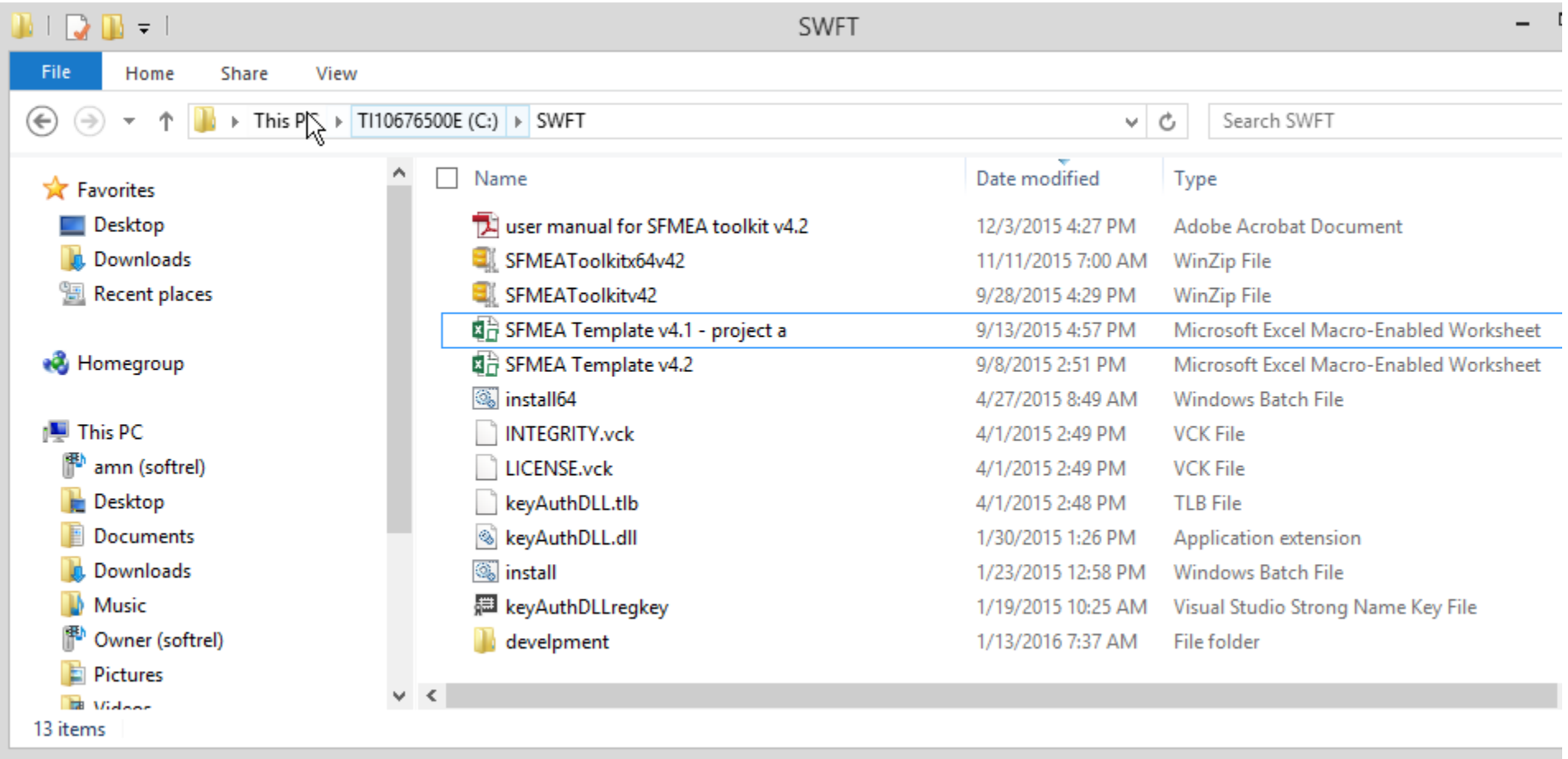
Opening the toolkit

- 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

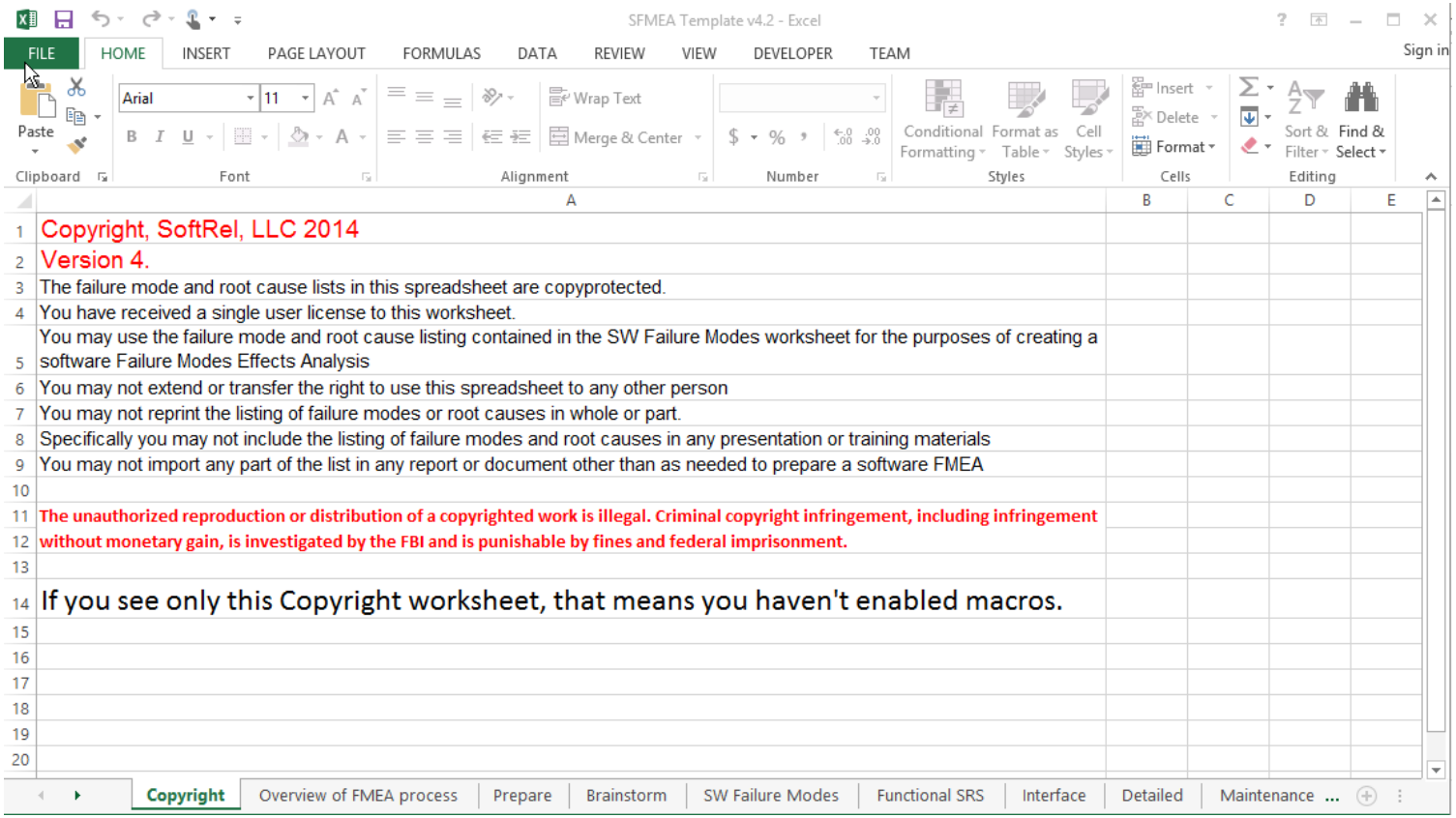


Copying the toolkit

- 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.
- As shown below the SFMEA Template v4.2 was copied to another template for “project a”.
- You can make as many templates as you like as long as they remain in the SWFT folder.

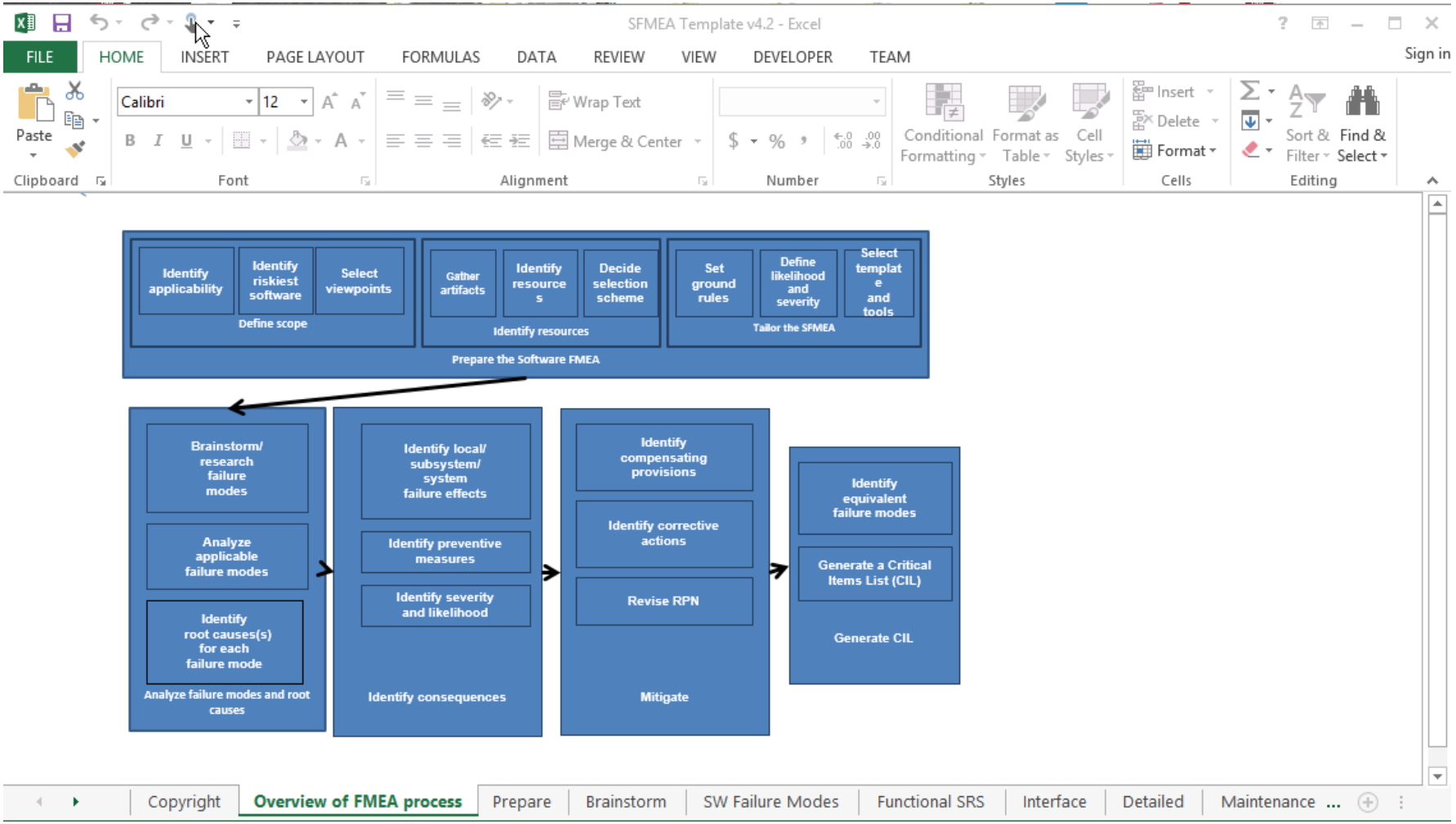


Copyright



- 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

Prepare the SFMEA

2.1 Define Scope

- Identify the applicability
- Identify riskiest parts of the software
- Identify most relevant viewpoints

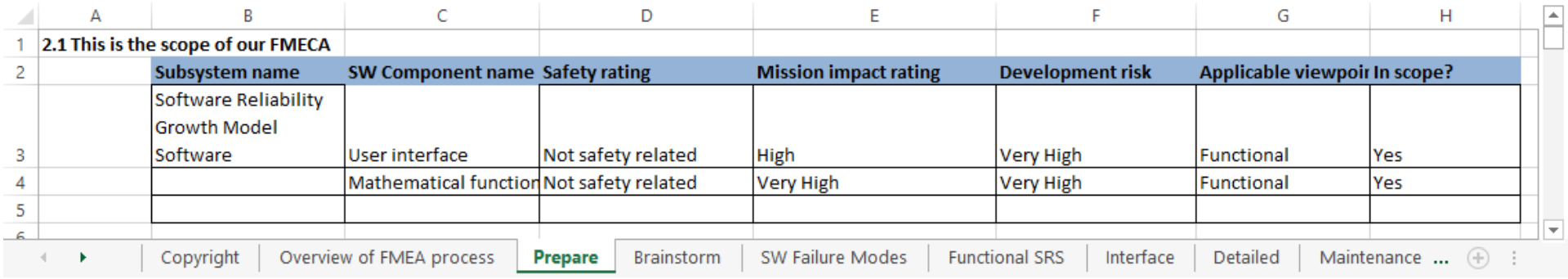
2.2 Identify resources

- Gather artifacts
- Identify the right people
- Decide selection scheme

2.3 Tailor the SFMEA

- Set ground rules
- Define likelihood and severity
- Select template and tools

2.1 Identify the scope



	A	B	C	D	E	F	G	H
1	2.1 This is the scope of our FMECA							
2		Subsystem name	SW Component name	Safety rating	Mission impact rating	Development risk	Applicable viewpoir	In scope?
3		Software Reliability Growth Model Software	User interface	Not safety related	High	Very High	Functional	Yes
4			Mathematical function	Not safety related	Very High	Very High	Functional	Yes
5								

Copyright | Overview of FMEA process | **Prepare** | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance ...

- Identify all of the software components in the system
- Identify the safety rating of each component
- Identify the mission impact of each component
- Identify the development risk – how problematic the particular code has been or is expected to be
- Identify the applicable viewpoints for each component. The choices are functional, interface, detailed, maintenance, usability, serviceability, vulnerability and production
- Identify which components are in scope for this SFMEA and which ones are not

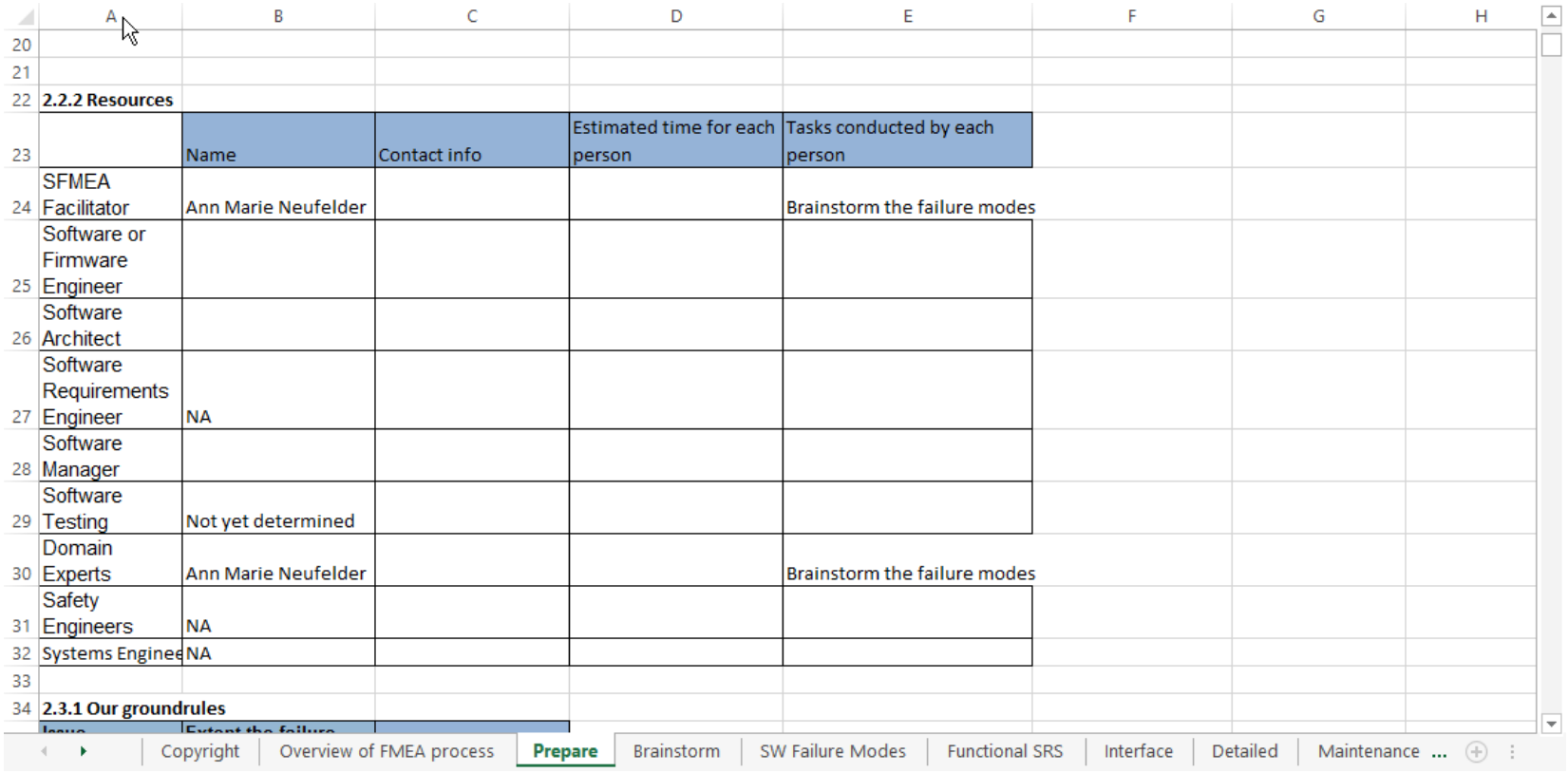
Identify resources

- Depending on the viewpoint selected, different artifacts are required for the analysis. Highlight the required artifacts from the below table. As shown here, either the System Requirements Spec or the Software requirements Spec is required for the functional SFMEA.

	A	B	C	D	E	F	G	H	
7	**2.2.1 Artifacts**		**Functional**	**Interface**	**Detailed and Vulnerability**	**Maintenance**	**Usability**	**Serviceability**	**A**
8		Systems Requirements Spec	One of these is required.				Required		
9		**Software Requirements Spec**	The SRS is preferred over the SyRS.						
10		System Architecture Design	Highly recommended						
11		Interface Control Spec (ICS), Interface Control Document (ICD), Interface Design Document (IDD)		At least one is required		IDD is highly recommended			
12		Software Detailed Design			One of these is required	Recommended			
13		Code				Required			
14		User interface (UI) design document		If the UI is in scope then this is required	If the code is related to the UI then this is required	If the change is related to the UI then this is required	Required		
15		User manuals or Help files or Use cases.			Required for vulnerability		Required		
16		Field reports, list of changes	Recommended			Required	Required		
17		Use Cases	Highly recommended				Required		
18		Software test plan/ procedures	May be required for corrective action			Required			
19		Installation scripts and guide						Required	

Copyright | Overview of FMEA process | **Prepare** | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance ...

Identify the right people



	A	B	C	D	E	F	G	H
20								
21								
22	**2.2.2 Resources**							
23		Name	Contact info	Estimated time for each person	Tasks conducted by each person			
24	SFMEA Facilitator	Ann Marie Neufelder			Brainstorm the failure modes			
25	Software or Firmware Engineer							
26	Software Architect							
27	Software Requirements Engineer	NA						
28	Software Manager							
29	Software Testing	Not yet determined						
30	Domain Experts	Ann Marie Neufelder			Brainstorm the failure modes			
31	Safety Engineers	NA						
32	Systems Enginee	NA						
33								
34	**2.3.1 Our groundrules**							

Copyright | Overview of FMEA process | **Prepare** | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance ...

- Identify who will be performing the SFMEA. Ensure that there are appropriate subject matter experts for the selected viewpoints. For example, the detailed, maintenance and vulnerability viewpoints require at least one software engineer to be involved with the SFMEA construction.

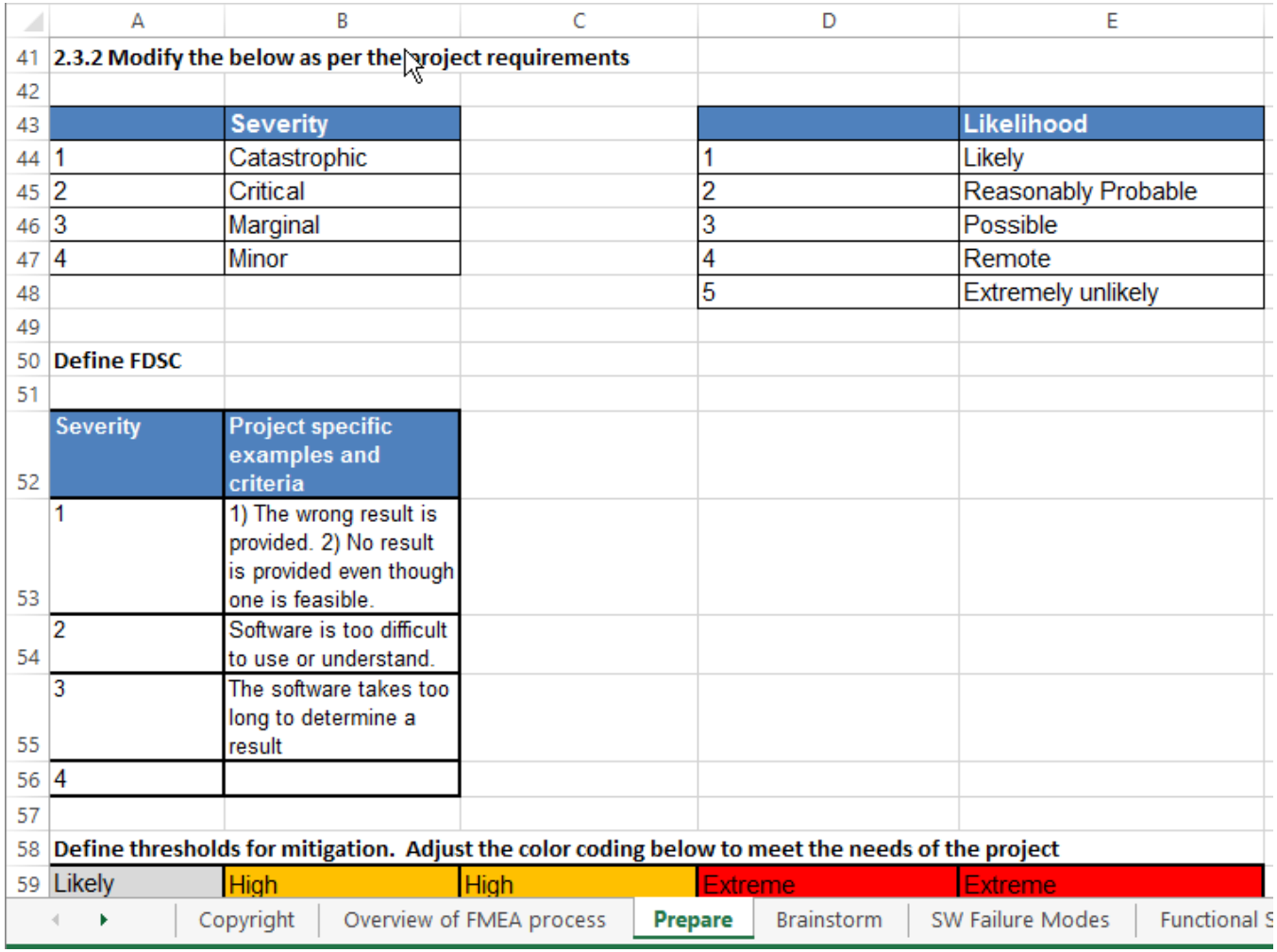
Set the ground rules

	A	B	C
34	**2.3.1 Our groundrules**		
35	**Issue**	**Extent the failure mode is propagated**	**Our decision**
36	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.	**Include**
37	Chain of interfaces	How many interface chains will we consider in one SFMEA row?	**Not applicable**
38	Network availability	Decide whether to assume that any network required for the system is available.	**Not applicable**
39	Speed and throughput	Decide whether to assume that the system is performing at maximum, typical or minimum speed and throughput.	**Typical**
40			

Copyright | Overview of FMEA process | **Prepare** | Bra

- Review the ground rules and make decisions for this SFMEA with regards to consideration of
  - human error
  - interface chains
  - network availability
  - speed/throughput.

Identify severity and likelihood



	A	B	C	D	E
41	**2.3.2 Modify the below as per the project requirements**				
42					
43		**Severity**			**Likelihood**
44	1	Catastrophic		1	Likely
45	2	Critical		2	Reasonably Probable
46	3	Marginal		3	Possible
47	4	Minor		4	Remote
48				5	Extremely unlikely
49					
50	**Define FDSC**				
51					
52	**Severity**	**Project specific examples and criteria**			
53	1	1) The wrong result is provided. 2) No result is provided even though one is feasible.			
54	2	Software is too difficult to use or understand.			
55	3	The software takes too long to determine a result			
56	4				
57					
58	**Define thresholds for mitigation. Adjust the color coding below to meet the needs of the project**				
59	Likely	High	High	Extreme	Extreme

Copyright | Overview of FMEA process | **Prepare** | Brainstorm | SW Failure Modes | Functional S

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

Import the artifacts into the template

	A	B	C	D	E	F	G	H
68	• The user will be presented with a graphical user interface consisting of four tabs to (i) select analyze and filter data, (ii) set-up and apply models, (iii) query model results, and (iv) evaluate models.							
69	• The first tab (select analyze and filter data) allows the user to:							
70	o Specify an input file with either inter-failure, failure time, or failure count data in Excel or CSV format.							
71	o Plot the data as time between failures, failure rate, or cumulative failures by selecting one of these options from a combo box.							
72	o Execute test such as the Laplace trend test and running arithmetic average to assess if the dataset exhibits reliability growth.							
73	o Save plots in various image file formats by clicking a button and specifying a name within a file dialog box.							
74	• The second tab (set-up and apply models) allows the user to:							
75	o Select a subset of the data to which models will be applied.							
76	o Indicate the prefix of the data that will be used to estimate parameters.							
77	o Select one or more models (Jelinski-Moranda, geometric, exponential, Yamada delayed S-shaped, and Weibull) from a list and estimate their parameters by clicking a button.							
78	o Plot the data and model fits as time between failure, failure rate, or cumulative failures by selecting one of these options from a combo box.							
79	• The third tab (query model results) allows the user to:							
80	o Estimate the time required to observe k additional failures.							
81	o Estimate the number of failures that would be observed given an additional amount of testing time.							
82	o Estimate the additional testing time required to achieve a desired reliability given a fixed mission time.							
83	• The fourth tab (evaluate models) allows the user to:							
84	o Apply goodness of fit measures such as the Akaike information criterion (AIC) and predictive sum of squares error (PSSE).							
85	o Rank models in a table according to their performance on goodness of fit measures, while also reporting raw numerical values of these measures.							
86								

Copyright | Overview of FMEA process | **Prepare** | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance ...

The SFMEA process is much easier when the artifacts are copied or imported into the template. In the above example, the SRS and Software architecture design is needed for the functional FMEA. These statements (and even pictures) should be copied in. Bold the requirements that are in scope for the SFMEA.

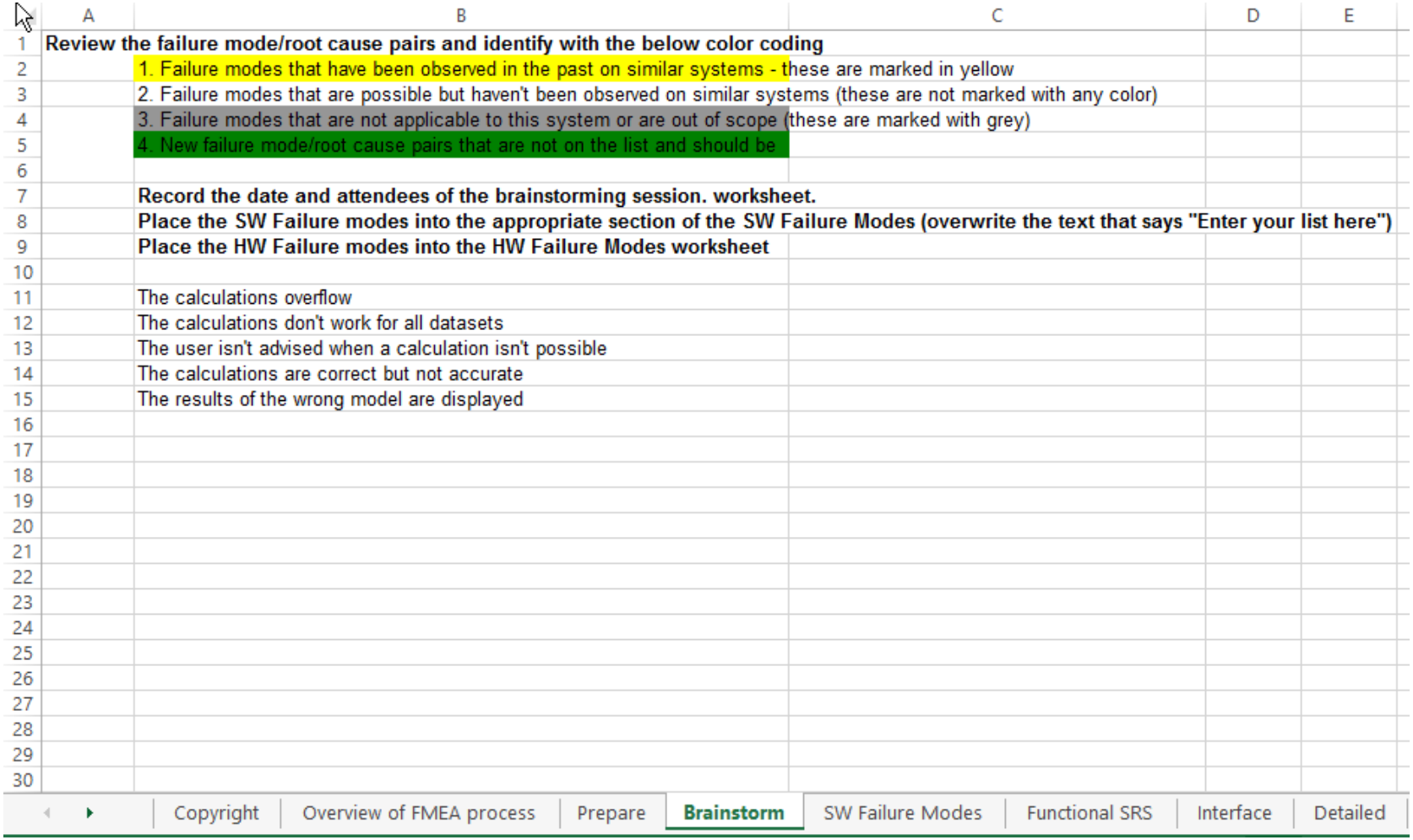
Step 3. Analyze failure modes and root causes

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

Analyze failure modes and root causes

- Brainstorm failure modes
- Analyze failure modes and root causes for each of the in scope SFMEA viewpoints
  - Functional
  - Interface
  - Detailed
  - Maintenance
  - Usability
  - Serviceability
  - Vulnerability
  - Production
- Your toolkit is populated with hundreds of failure modes and root causes

Brainstorm Failure modes

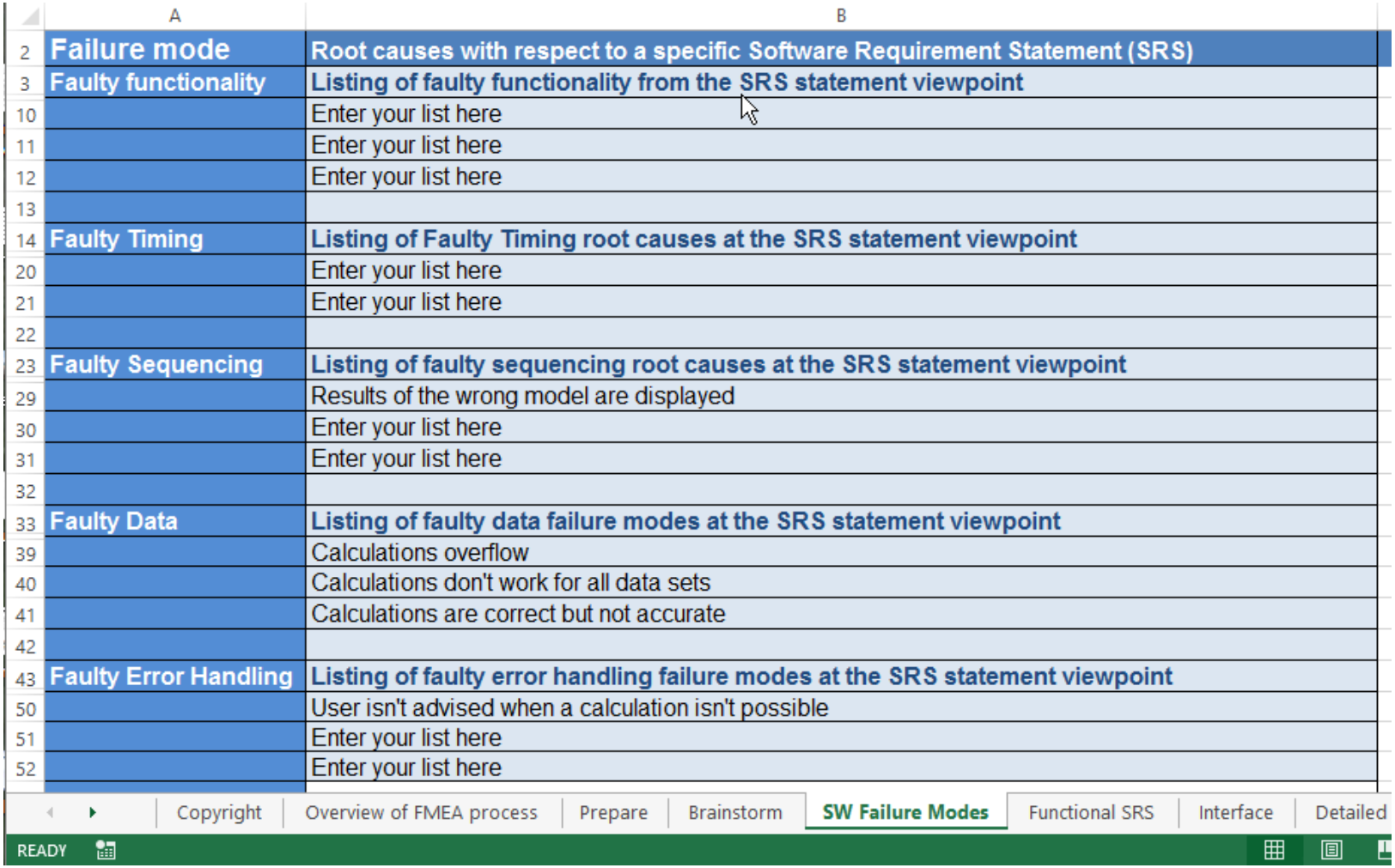


	A	B	C	D	E
1	**Review the failure mode/root cause pairs and identify with the below color coding**				
2		1. Failure modes that have been observed in the past on similar systems - these are marked in yellow			
3		2. Failure modes that are possible but haven't been observed on similar systems (these are not marked with any color)			
4		3. Failure modes that are not applicable to this system or are out of scope (these are marked with grey)			
5		4. New failure mode/root cause pairs that are not on the list and should be			
6					
7		**Record the date and attendees of the brainstorming session. worksheet.**			
8		**Place the SW Failure modes into the appropriate section of the SW Failure Modes (overwrite the text that says "Enter your list here")**			
9		**Place the HW Failure modes into the HW Failure Modes worksheet**			
10					
11		The calculations overflow			
12		The calculations don't work for all datasets			
13		The user isn't advised when a calculation isn't possible			
14		The calculations are correct but not accurate			
15		The results of the wrong model are displayed			
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

Copyright | Overview of FMEA process | Prepare | **Brainstorm** | SW Failure Modes | Functional SRS | Interface | Detailed

- The toolkit is packaged with hundreds of failure mode/root cause pairs.
- However, you also have the ability to identify additional failure modes or root causes.

Brainstorm Failure Modes



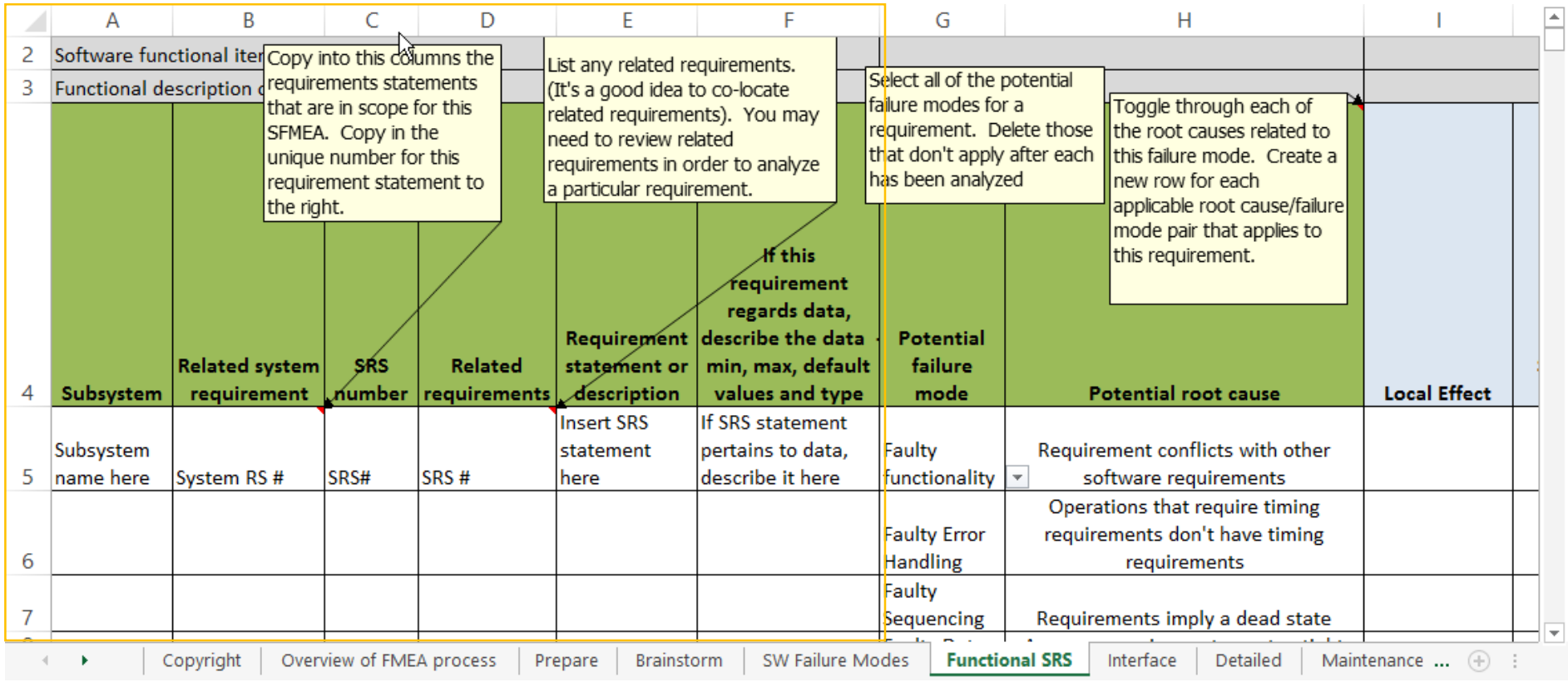
	A	B
2	**Failure mode**	**Root causes with respect to a specific Software Requirement Statement (SRS)**
3	**Faulty functionality**	**Listing of faulty functionality from the SRS statement viewpoint**
10		Enter your list here
11		Enter your list here
12		Enter your list here
13		
14	**Faulty Timing**	**Listing of Faulty Timing root causes at the SRS statement viewpoint**
20		Enter your list here
21		Enter your list here
22		
23	**Faulty Sequencing**	**Listing of faulty sequencing root causes at the SRS statement viewpoint**
29		Results of the wrong model are displayed
30		Enter your list here
31		Enter your list here
32		
33	**Faulty Data**	**Listing of faulty data failure modes at the SRS statement viewpoint**
39		Calculations overflow
40		Calculations don't work for all data sets
41		Calculations are correct but not accurate
42		
43	**Faulty Error Handling**	**Listing of faulty error handling failure modes at the SRS statement viewpoint**
50		User isn't advised when a calculation isn't possible
51		Enter your list here
52		Enter your list here

Copyright | Overview of FMEA process | Prepare | Brainstorm | **SW Failure Modes** | Functional SRS | Interface | Detailed

READY

Once the failure modes and root causes are brainstormed they can be typed directly into the SW Failure Modes worksheet. This will allow those user defined failure modes/root causes to be included in the pull-down menus.

Functional (SRS) SFMEA viewpoint



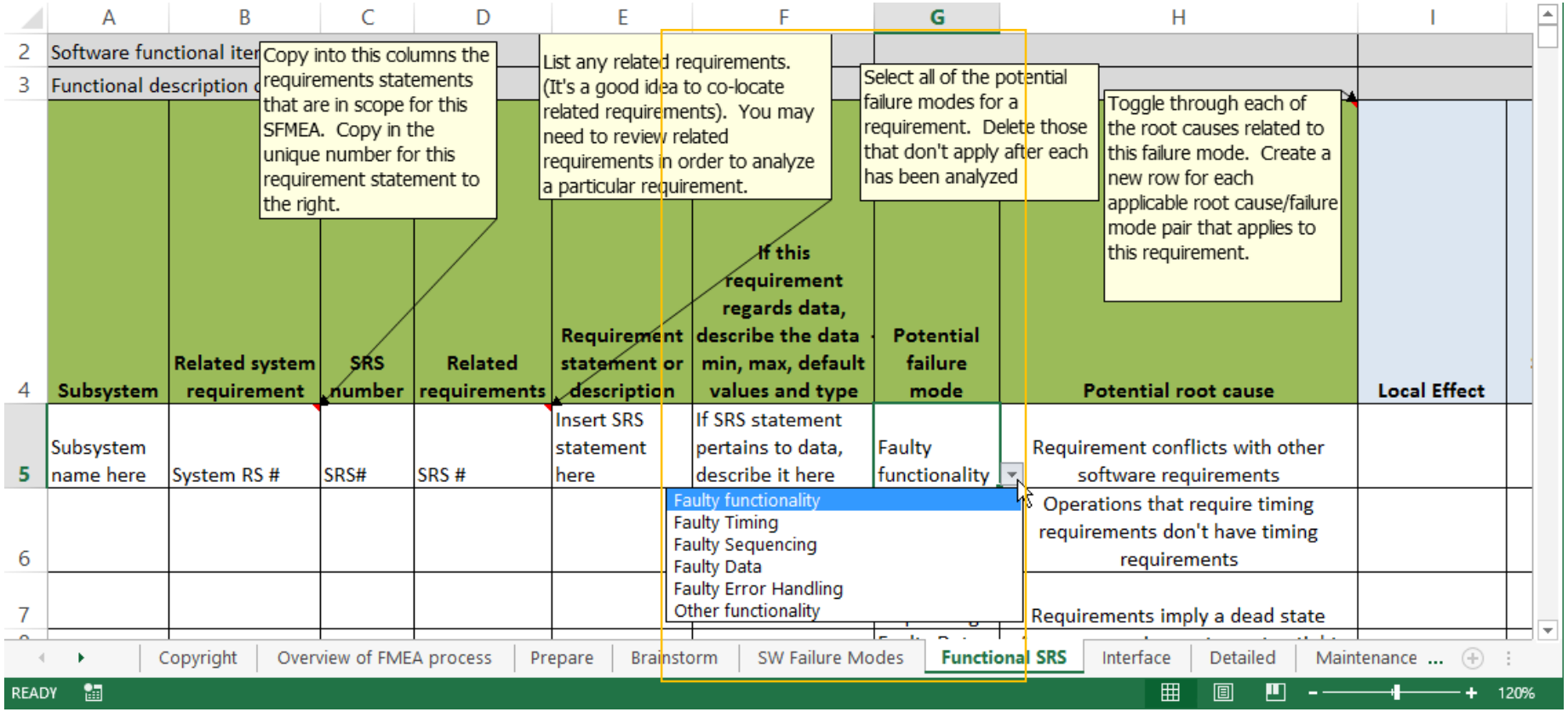
The Functional SRS worksheet is pre-populated with a template.

There are “call outs” to guide you in setting up the SFMEA.

First, copy in all of the in scope software requirements statements so that each one has it’s own area.

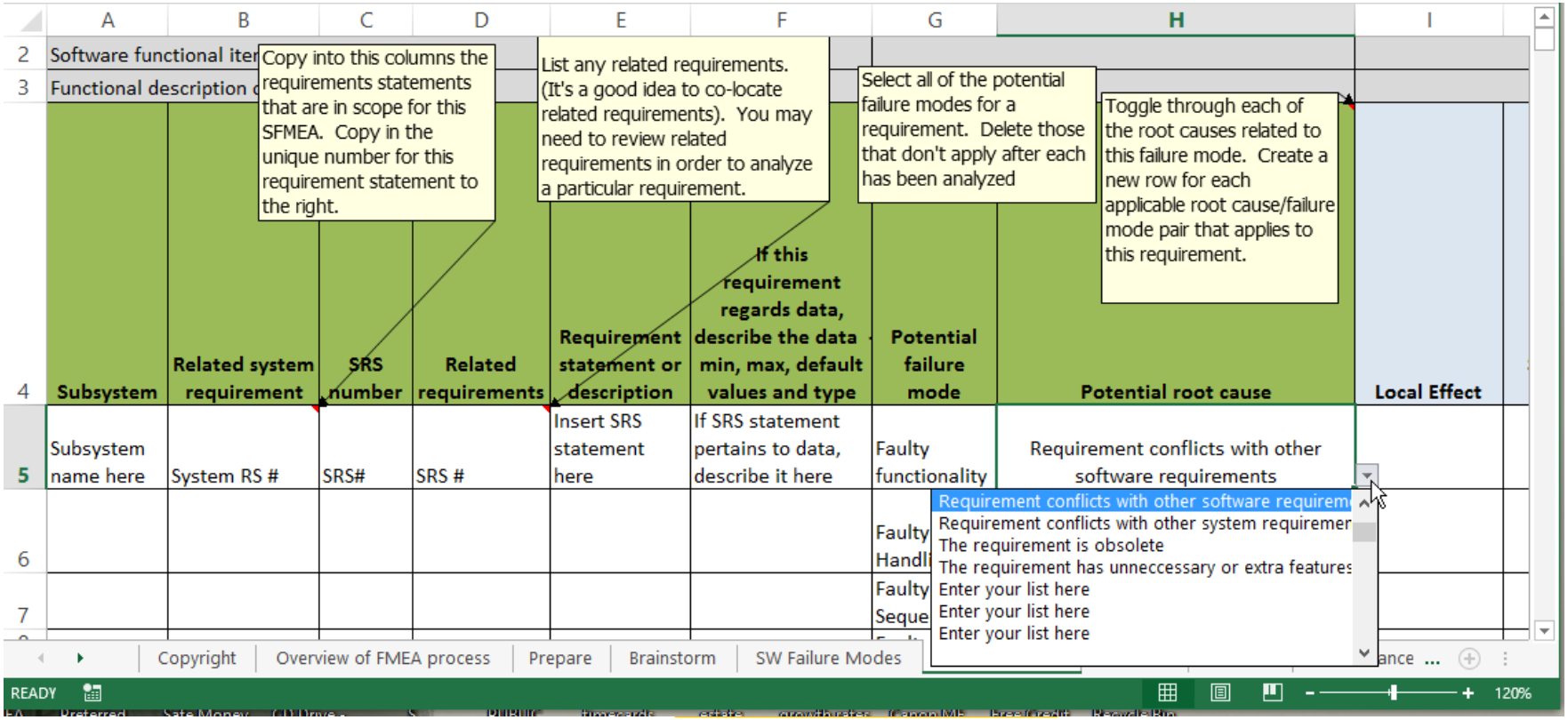
Any related requirements are also copied in.

Functional (SRS) SFMEA viewpoint



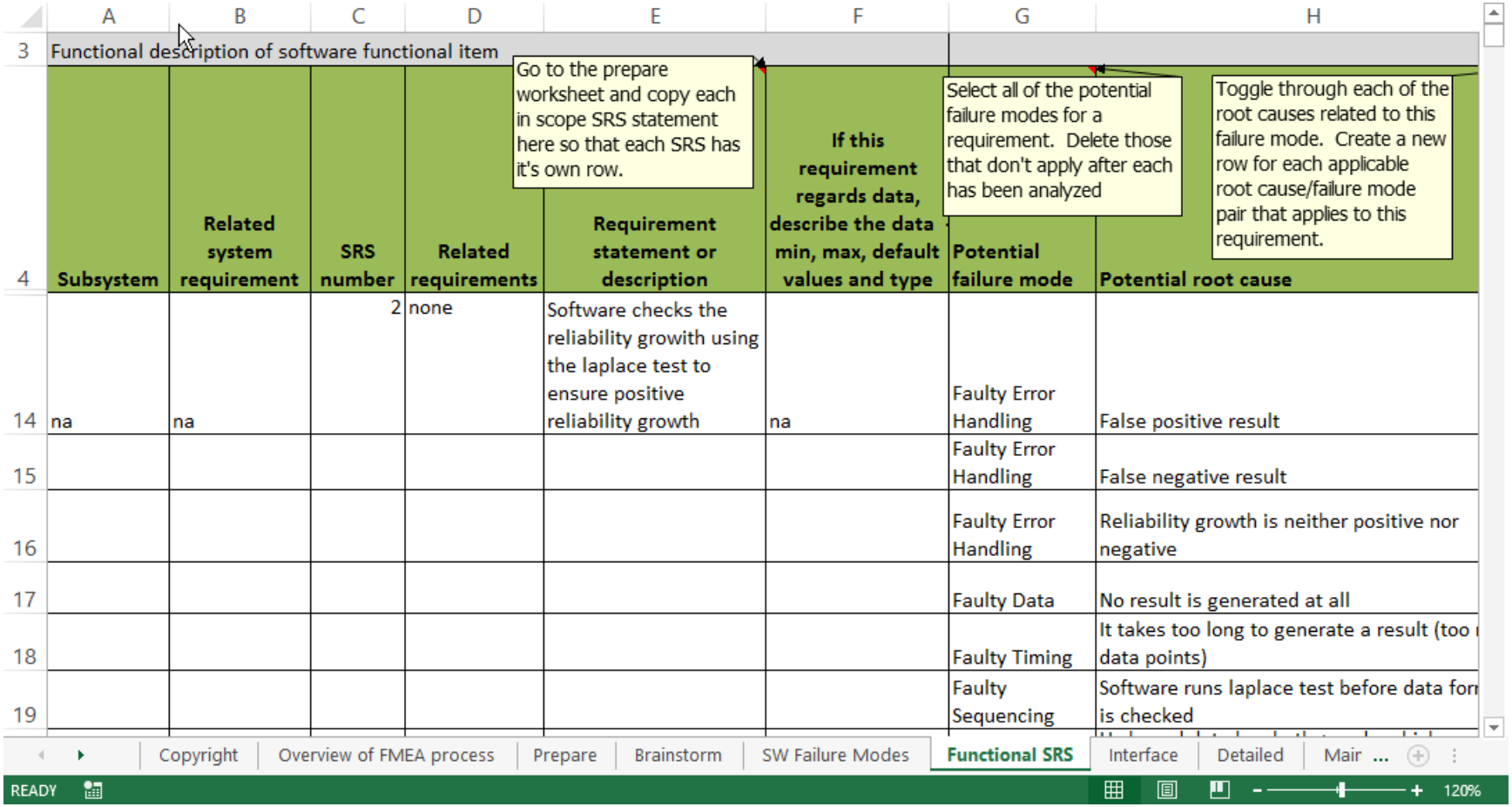
In the event that you don't wish to use the built in template, the failure modes are selectable with the pulldown menu.

Functional (SRS) SFMEA viewpoint



The root causes are also selectable with a pull down menu. If you add root causes in the Brainstorm worksheet these will appear in the pulldown menus.

Functional SRS Example



	A	B	C	D	E	F	G	H
3	Functional description of software functional item							
4	**Subsystem**	**Related system requirement**	**SRS number**	**Related requirements**	**Requirement statement or description**	**If this requirement regards data, describe the data min, max, default values and type**	**Potential failure mode**	**Potential root cause**
14	na	na	2	none	Software checks the reliability growth using the laplace test to ensure positive reliability growth	na	Faulty Error Handling	False positive result
15							Faulty Error Handling	False negative result
16							Faulty Error Handling	Reliability growth is neither positive nor negative
17							Faulty Data	No result is generated at all
18							Faulty Timing	It takes too long to generate a result (too data points)
19							Faulty Sequencing	Software runs laplace test before data for is checked

Go to the prepare worksheet and copy each in scope SRS statement here so that each SRS has it's own row.

Select all of the potential failure modes for a requirement. Delete those that don't apply after each has been analyzed

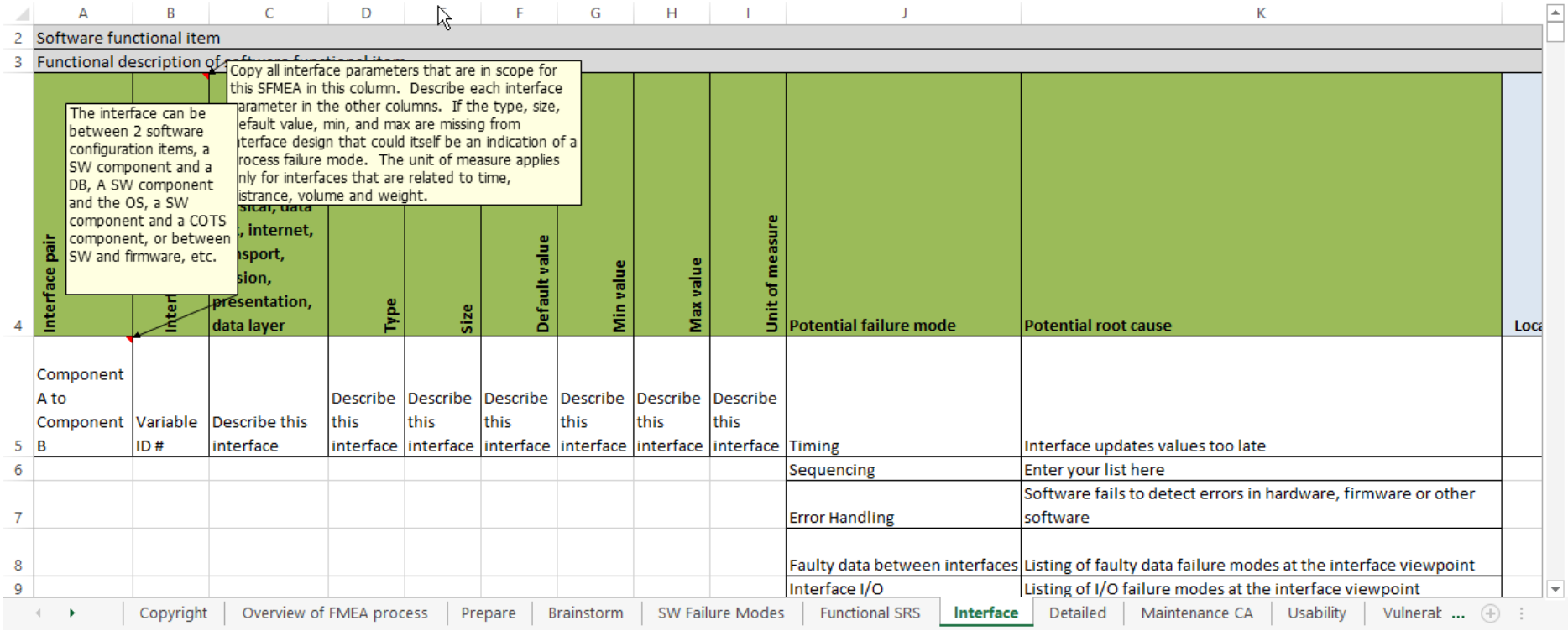
Toggle through each of the root causes related to this failure mode. Create a new row for each applicable root cause/failure mode pair that applies to this requirement.

Copyright | Overview of FMEA process | Prepare | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Mair ...

READY 120%

This example shows one requirement and the associated failure modes and root causes. Each in scope SRS statement would have a section similar to the above.

Interface SFMEA



	A	B	C	D	E	F	G	H	I	J	K
2	Software functional item										
3	Functional description of software functional item										
4	Interface pair	Inter	, internet, sport, sion, presentation, data layer	Type	Size	Default value	Min value	Max value	Unit of measure	Potential failure mode	Potential root cause
5	Component A to Component B	Variable ID #	Describe this interface	Describe this interface	Describe this interface	Describe this interface	Describe this interface	Describe this interface	Describe this interface	Timing	Interface updates values too late
6										Sequencing	Enter your list here
7										Error Handling	Software fails to detect errors in hardware, firmware or other software
8										Faulty data between interfaces	Listing of faulty data failure modes at the interface viewpoint
9										Interface I/O	Listing of I/O failure modes at the interface viewpoint

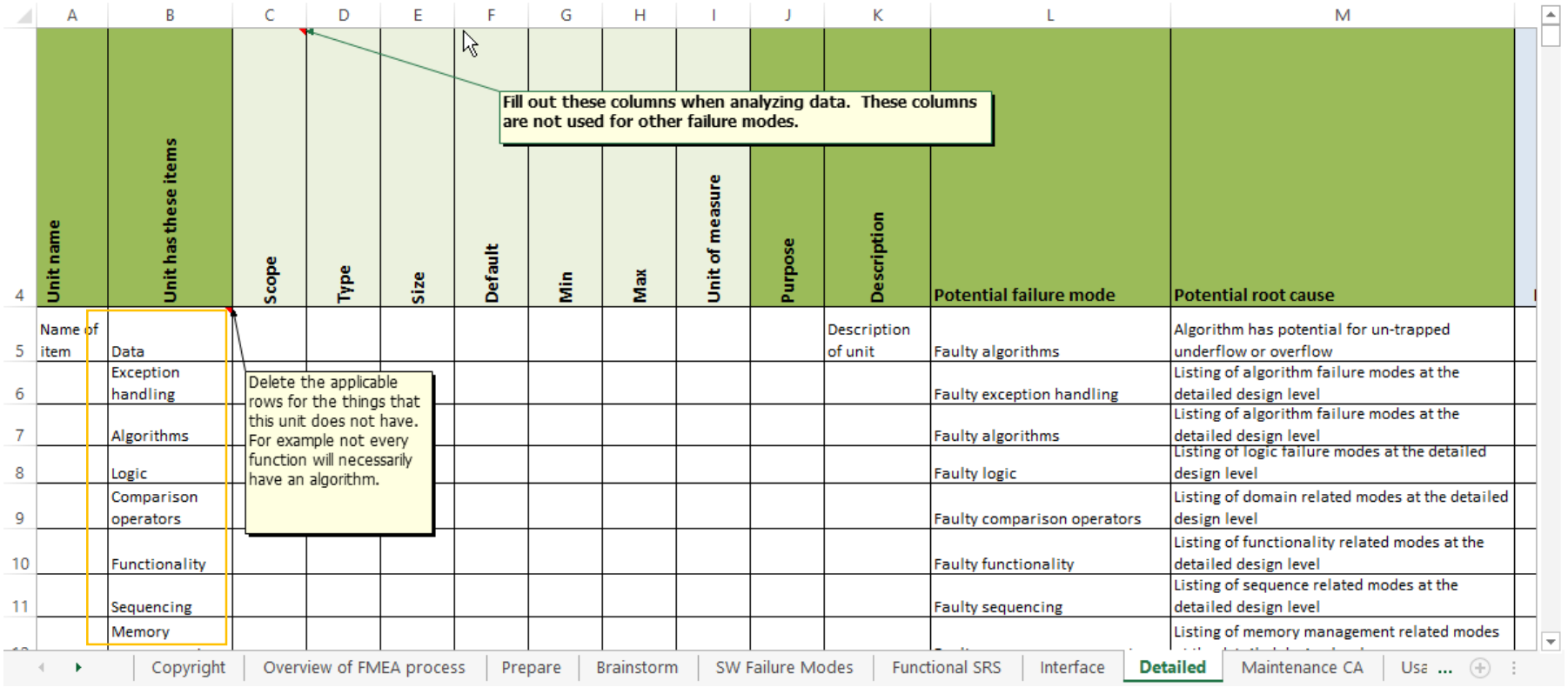
Copy all interface parameters that are in scope for this SFMEA in this column. Describe each interface parameter in the other columns. If the type, size, default value, min, and max are missing from interface design that could itself be an indication of a process failure mode. The unit of measure applies only for interfaces that are related to time, distrance, volume and weight.

The interface can be between 2 software configuration items, a SW component and a DB, A SW component and the OS, a SW component and a COTS component, or between SW and firmware, etc.

Copyright | Overview of FMEA process | Prepare | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usability | Vulnerab ...

- The Interface SFMEA viewpoint has a slightly different template than the functional SFMEA since it is focused on the interface between 2 software components or between a software/hardware component
- From the interface design specification enter in the variable ID, type of interface, type size, default value, minimum value, maximum value and applicable unit of measure.
- If these items are not in the IDS that, in itself, can indicate a potential failure mode.
- The Interface SFMEA has it’s own set of built in failure modes and root causes that apply to the interface viewpoint.

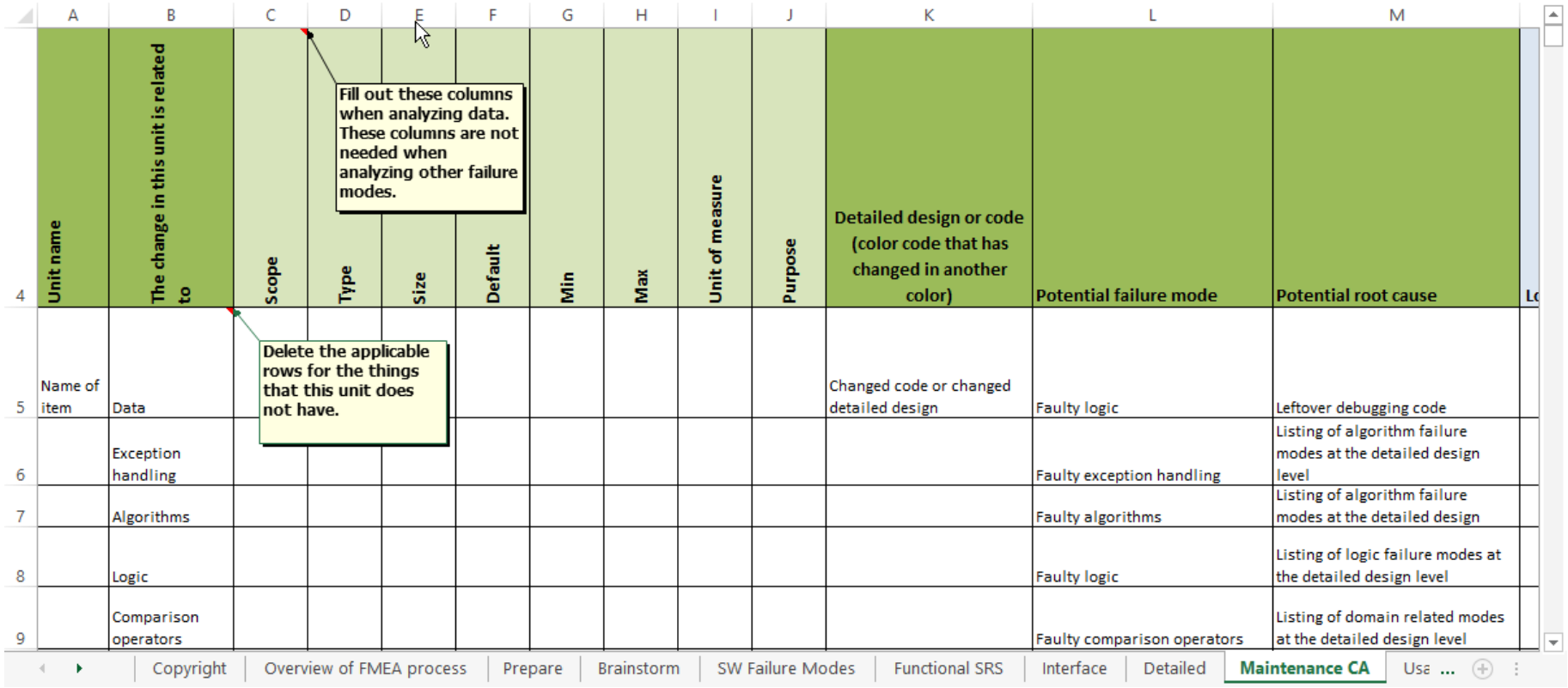
Detailed SFMEA



Unit name	Unit has these items	Scope	Type	Size	Default	Min	Max	Unit of measure	Purpose	Description	Potential failure mode	Potential root cause
Name of item	Data									Description of unit	Faulty algorithms	Algorithm has potential for un-trapped underflow or overflow
	Exception handling										Faulty exception handling	Listing of algorithm failure modes at the detailed design level
	Algorithms										Faulty algorithms	Listing of algorithm failure modes at the detailed design level
	Logic										Faulty logic	Listing of logic failure modes at the detailed design level
	Comparison operators										Faulty comparison operators	Listing of domain related modes at the detailed design level
	Functionality										Faulty functionality	Listing of functionality related modes at the detailed design level
	Sequencing										Faulty sequencing	Listing of sequence related modes at the detailed design level
	Memory											Listing of memory management related modes

- The Detailed SFMEA viewpoint has a slightly different template than the functional SFMEA since it is focused on the detailed design of a particular component.
- First the analyst needs to identify what is relevant for this particular function, module or class. Data, exception handling, functionality and memory are always relevant. A function may or may not have logic, algorithms, comparison operators, or sequences.
- For each particular function, delete the characteristics that don't apply to that function. Then explore the failure modes and root causes that pertain to the relevant characteristics of the function.

Maintenance SFMEA



	A	B	C	D	E	F	G	H	I	J	K	L	M
4	Unit name	The change in this unit is related to	Scope	Type	Size	Default	Min	Max	Unit of measure	Purpose	Detailed design or code (color code that has changed in another color)	Potential failure mode	Potential root cause
5	Name of item	Data									Changed code or changed detailed design	Faulty logic	Leftover debugging code
6		Exception handling										Faulty exception handling	Listing of algorithm failure modes at the detailed design level
7		Algorithms										Faulty algorithms	Listing of algorithm failure modes at the detailed design
8		Logic										Faulty logic	Listing of logic failure modes at the detailed design level
9		Comparison operators										Faulty comparison operators	Listing of domain related modes at the detailed design level

Fill out these columns when analyzing data. These columns are not needed when analyzing other failure modes.

Delete the applicable rows for the things that this unit does not have.

Copyright | Overview of FMEA process | Prepare | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usa ...

- The Maintenance SFMEA template is exactly like the detailed SFMEA template.
- Except that the focus is on the detailed design or code that has CHANGED since the last established baseline.
- It's best to copy in the detailed design or code and highlight the changes in another color or font. Then analyze what can go wrong with the change.

Usability SFMEA

	A	B	C	D	E	F	G
2	Software functional item						
3	Functional description of software functional item						
4	**Subsystem**	**Related system requirement**	**SRS number**	**Related requirements**	**Use case**	**Potential failure mode**	**Potential root cause**
5						Faulty assumptions about end users	User documents are obsolete
6						Overly cumbersome software	Listing of overly cumbersome root causes
7						Software isn't robust for common human errors	Listing of robustness root causes
8						Faulty assumptions about end users	Listing of root causes for faulty assumptions about the end user
9						Legal users use the software for the wrong purpose	Listing of root causes for legal users using the software for the wrong purposes
10						Legal users have access they shouldn't have	Listing of root causes for legal users having access that they shouldn't have
11							
12							
13							
14							
15							
16							

Import each of the use cases and the relevant requirements and related requirements

... | Overview of FMEA process | Prepare | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usabil ...

- The usability SFMEA focuses on the use cases and what can go wrong when there are humans interfacing with the software.
- Copy in each of the in scope use cases and analyze each one, one at a time. The template has pre-populated failure modes and root causes.
- Delete the failure modes and root causes that aren't relevant to a particular use case. Analyze the remainder.

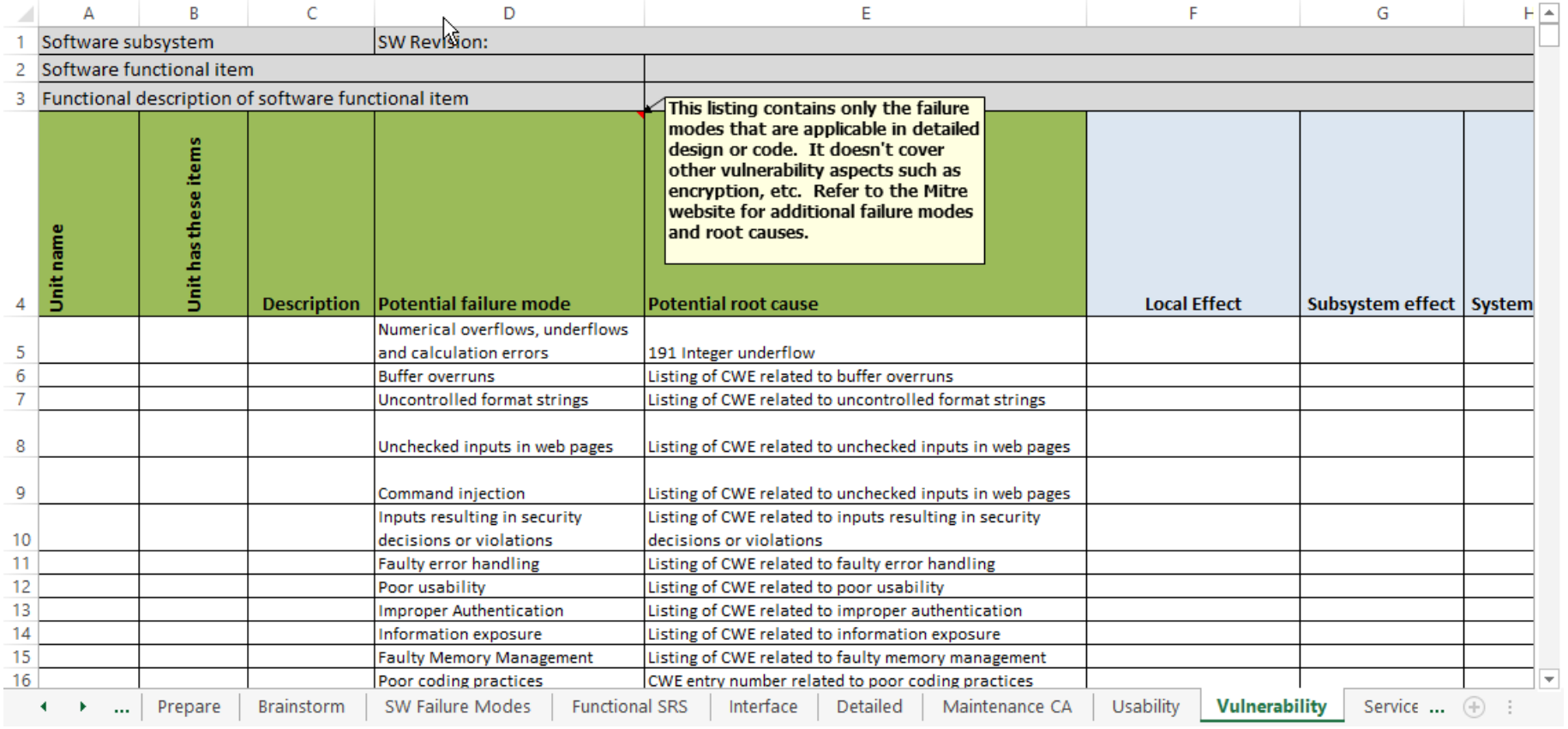
Serviceability SFMEA

	A	B	C	D	E	F	G	H
1	Software subsystem					SW Revision:		
2	Software functional item							
3	Functional description of software functional item							
4	**Version**	**Installation script**	**Potential failure mode**	**Potential root cause**	**Local Effect**	**Subsystem effect**	**System effect**	**Preventive measures**
5			Insufficient personnel or resources to service software	Insufficient documentation for service packs				
6			Faulty installation scripts for update and install	Installs when permissions aren't sufficient				
7								

... | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usability | Vulnerability | **Serviceability** | F ...

- The serviceability SFMEA focuses on the installation and upgrades of the software. This can be particular critical for software that is mass deployed or software that is deployed to geographically difficult to reach areas.
- The two basic reasons that installations or upgrades fail is
  - The installation is too difficult for someone other than a software engineer to do
  - The installation has faulty install scripts
- There are numerous root causes for these failure modes which are contained in the pre-populated pull-down menus.

Vulnerability SFMEA



	A	B	C	D	E	F	G	H
1	Software subsystem			SW Revision:				
2	Software functional item							
3	Functional description of software functional item				This listing contains only the failure modes that are applicable in detailed design or code. It doesn't cover other vulnerability aspects such as encryption, etc. Refer to the Mitre website for additional failure modes and root causes.			
4	Unit name	Unit has these items	Description	Potential failure mode	Potential root cause	Local Effect	Subsystem effect	System
5				Numerical overflows, underflows and calculation errors	191 Integer underflow			
6				Buffer overruns	Listing of CWE related to buffer overruns			
7				Uncontrolled format strings	Listing of CWE related to uncontrolled format strings			
8				Unchecked inputs in web pages	Listing of CWE related to unchecked inputs in web pages			
9				Command injection	Listing of CWE related to unchecked inputs in web pages			
10				Inputs resulting in security decisions or violations	Listing of CWE related to inputs resulting in security decisions or violations			
11				Faulty error handling	Listing of CWE related to faulty error handling			
12				Poor usability	Listing of CWE related to poor usability			
13				Improper Authentication	Listing of CWE related to improper authentication			
14				Information exposure	Listing of CWE related to information exposure			
15				Faulty Memory Management	Listing of CWE related to faulty memory management			
16				Poor coding practices	CWE entry number related to poor coding practices			

... | Prepare | Brainstorm | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usability | Vulnerability | Service ...

- The vulnerability SFMEA focuses on the detailed design and code as well as use cases. Note that there are many other vulnerabilities. However, this SFMEA focuses on those that are related to the design or code.
- The design/code related vulnerability related failure modes are listed. There are many Common Weakness Entries (CWE) for each of the failure modes.

Production SFMEA

	A	B	C	D	E	F
1	Software subsystem					SW Revision:
2	Software functional item					
3	Functional description of software functional item					
4	**Software project/ release**	**Description**	**Potential failure mode**	**Potential root cause**	**Local Effect**	**Subsystem effect**
5			Faulty scheduling practices	The size of the software is grossly underestimated		
6			Faulty personnel staffing	Software engineers are hired based on knowledge of language instead of knowledge of software engineering		
7			Faulty requirements analysis practices	Listing of root causes for faulty requirements analysis practices		
8			Faulty design practices	Listing of root causes for faulty design practices		
9			Faulty implementation practices	Listing of root causes for faulty implementation practices		
10			Faulty testing practices	Listing of root causes for faulty testing practices		
11			Faulty defect prevention practices	Listing of root causes for faulty defect prevention practices		
12			Faulty tool selection	Listing of root causes for faulty tool selection		
13			Faulty change control practices	Listing of root causes for faulty change control		
14						
15						
16						
17						
18						
19						
20						
21						
22						

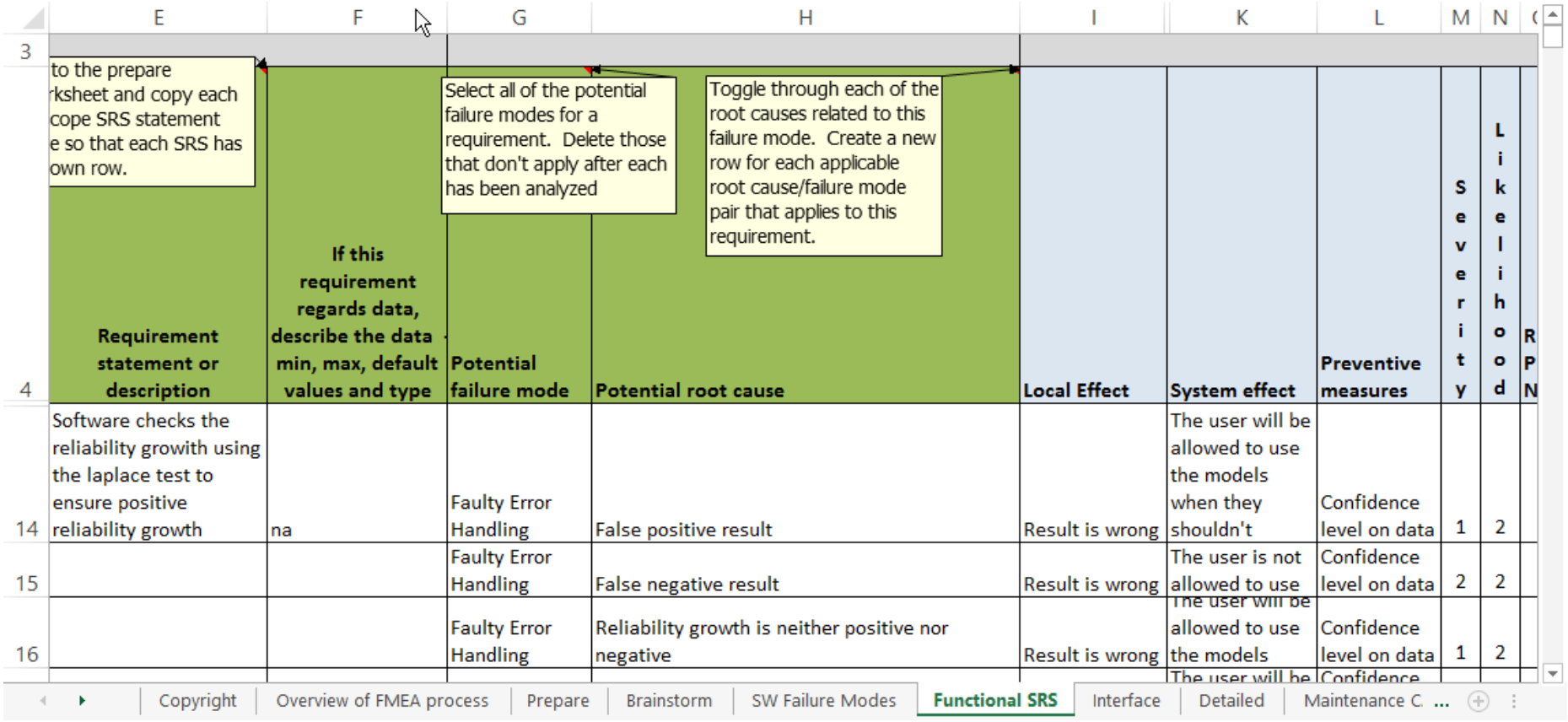
... | SW Failure Modes | Functional SRS | Interface | Detailed | Maintenance CA | Usability | Vulnerability | Serviceability | **Production** | C ...

- The Production SFMEA is the only viewpoint that focuses on the processes that produce the software as opposed to the software product itself.
- The key failure modes related to production, such as faulty scheduling and staffing are listed as well as numerous root causes for each failure mode.

Step 4. Identify Consequences

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

Identify Consequences



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.

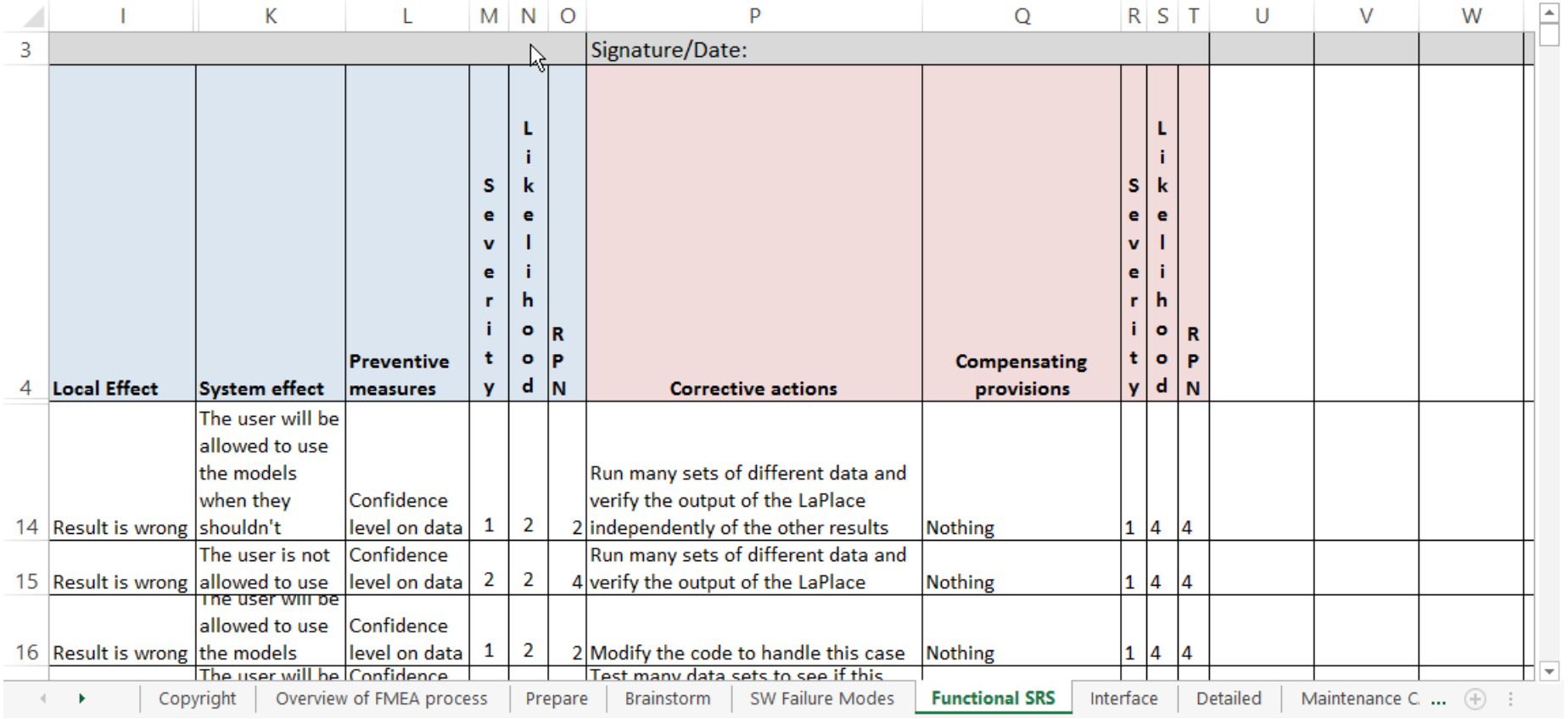
Tip: It's usually best to analyze all of the effects and preventive measures first and then analyze the severity and likelihood in one last pass.

The Risk Probability Number (RPN) is automatically calculated.

Step 5. Identify Mitigations

Identify corrective actions, compensating provisions and revised RPN

Identify Mitigations



	I	K	L	M	N	O	P	Q	R	S	T	U	V	W
3							Signature/Date:							
4	**Local Effect**	**System effect**	**Preventive measures**	**Severity**	**Likelihood**	**RPN**	**Corrective actions**	**Compensating provisions**	**Severity**	**Likelihood**	**RPN**			
14	Result is wrong	The user will be allowed to use the models when they shouldn't	Confidence level on data	1	2	2	Run many sets of different data and verify the output of the LaPlace independently of the other results	Nothing	1	4	4			
15	Result is wrong	The user is not allowed to use	Confidence level on data	2	2	4	Run many sets of different data and verify the output of the LaPlace	Nothing	1	4	4			
16	Result is wrong	The user will be allowed to use the models	Confidence level on data	1	2	2	Modify the code to handle this case	Nothing	1	4	4			
		The user will be	Confidence				Test many data sets to see if this							

Copyright | Overview of FMEA process | Prepare | Brainstorm | SW Failure Modes | **Functional SRS** | Interface | Detailed | Maintenance C. ...

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.