Mapping of classes to IEEE 1633 Recommended Practices for Software Reliability

This table shows which of the Softrel courses map to each of the sections of the recommended practices.

Training class	Software reliability task	
IEEE 1633 Recommended Practices for software reliability	5.1. Planning for software reliability	
	5.1.1	Characterize the software system
Software Failure Modes Effects Analysis Toolkit	5.2. Develop Failure Modes Model	
	5.2.1	Perform a defect root cause analysis
	5.2.2	Perform Software Failure Modes Effects Analysis (SFMEA)
	5.2.3	Include Software in the System Fault Tree Analysis
	5.3. Apply software reliability during development	
Integrating software and hardware reliability	5.3.1	Identify/Obtain the initial system reliability objective – The required or desired MTBF, failure rate, availability, reliability for the entire system.
IEEE 1633 Recommended Practices for software reliability	5.3.2	Perform a Software Reliability Assessment and Prediction – Predict the MTBF, failure rate, availability, reliability and confidence bounds for each software LRU.
	5.3.3	Sanity Check the Prediction – Compare the prediction to established ranges for similar software LRUs
Integrating software and hardware reliability	5.3.4	Merge the predictions into the overall system predictions – Various methods exist to combine the software and hardware predictions into one system prediction.
IEEE 1633 Recommended Practices for software reliability	5.3.5	Determine an appropriate overall Software Reliability Requirement – Now that the system prediction is complete, revisit the initial reliability objective and modify as needed.
	5.3.6	Plan the reliability growth – Compute the software predictions during and after a specific level of reliability growth testing.
	5.3.7	Perform a Sensitivity Analysis – Identify the practices, processes, techniques, risks that are most sensitive to the predictions. Perform tradeoffs.
Integrating software and hardware reliability	5.3.8	Allocate the Required Reliability to the Software LRUs– The software and hardware components are allocated a portion of the system objective based on the predicted values for each LRU.
	5. 4. Apply Software Reliability during Testing	
Design for test	5.4.1	Develop a reliability test suite
	5.4.3	Measure test coverage
Integrating software and hardware reliability	5.4.4	Collect Fault and Failure Data– This data is collected during testing and is required for using the
	5.4.5	Select Reliability Growth Models – The best models are those that fit the observed fault trends.
	5.5 Make a release decision	
	5.5.1	Determine release stability
	5.5.2	Forecast additional test duration
	5.5.3	Forecast remaining defects and effort required to fix them
	6.0 Software reliability models	
IEEE 1633 Recommended Practices for software reliability	6.1	Overview
	6.2	Models that can be used before testing
Integrating software and hardware predictions	6.3	Models that can be used during and after testing

IEEE Recommended Practices for Software Reliability 2 Day Training

Who should attend: Reliability engineers, systems engineers, software QA, software test engineers, software management, and acquisitions personnel. Class handouts include the software reliability toolkit.

1.0 Getting started

Greetings and Introductions
Software Reliability Timeline
Industry guidance available for software reliability
Vocabulary
Overview of models that predict and estimate software reliability models
Hard facts
Mapping software to hardware reliability

- Failure modes that do and do not apply
- Where software fits within the product lifecycle

Common myths

- Top list of things that everyone thinks is related to reliable software (but really isn't)
- Why software reliability growth is more limited than you think

Overview of methods for reliability testing

2.0 Planning for software reliability

Topics	**Section of IEEE 1633 2017**
Characterize the software system	5.1.1
Define failures and criticality	5.1.2
Perform an initial risk assessment	5.1.3

3.0 Apply software reliability during development

Section of this presentation	**Section of IEEE 1633 2017**
1. Predict normalized effective size	5.3.2.3.1
2. Predict testing or fielded defect density using the SEI CMMi, industry type, Shortcut Model	
3. Predict total testing and fielded defects	
4. Predict when defects will be discovered over time	5.3.2.3.2
5. Predict failure rate and MTTF	5.3.2.3.3
5.1. Sanity check the predictions	5.3.3
6. Predict reliability	5.3.2.3.4
7. Predict availability	5.3.2.3.5
8. Sensitivity analysis	5.3.7
9. Apply predictions with incremental development	5.3.2.4
10. Predict defect pileup	5.3.6
11. Predict staff required to maintain software	5.5
Detailed methods for steps 1-8	
Step 1. Predicting size of COTS components	5.3.2.5
Step 2. Advanced models for predicting defect density - Quick Assessment, Full-scale, Neufelder, Rome Laboratory, Historical Data	5.3.2.3.1, 6.2 and Annex B
Step 4. Other options for predicting growth rate	Not included
Step 8. Advanced sensitivity analysis	5.3.7

Integrating software and hardware predictions

Who should attend: Reliability engineers, systems engineers, software QA, software test engineers, software management and acquisitions personnel.

1.0 Combining software and hardware reliability

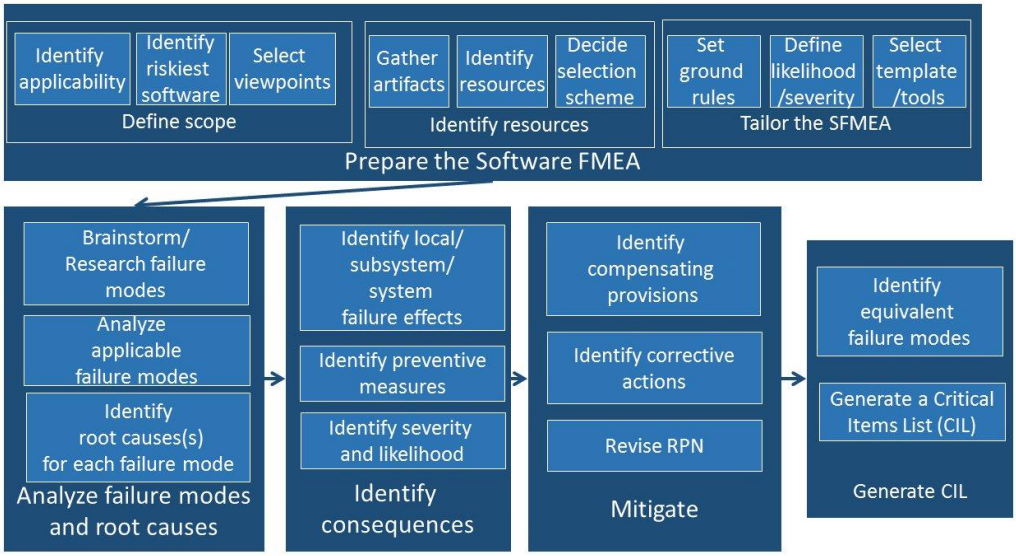
Section of this training	**Section of IEEE 1633 Recommended Practices for Software Reliability 2017**
1.0 Identify an initial system reliability objective	5.3.1
2.0 Determine an appropriate overall software reliability requirement	5.3.5
3.0 Merge the software reliability predictions into the system prediction	5.3.4
4.0 Allocate the required reliability to the software LRUs	5.3.8
5.0 If the system objective can't be met, perform a system level sensitivity analysis	5.3.7.2

2.0 Apply software reliability during testing

Section of this training	**Section of IEEE 1633 Recommended Practices for Software Reliability 2017**
Overview -The reliability growth curve for software	5.4.4
How to know where the program is on that curve	
1.0 Collect the data	5.4.4
2.0 Plot the data	5.4.4 and 5.4.5
How to estimate the failure rate, MTBF from that curve	
3.0 Select the best model for the current trend	5.4.5
4.0 Compute the reliability figures of merit	6.3 and Annex B
5.0 Validate the accuracy of the estimation	5.4.7
6.0 Make a release decision	5.5

Software Failure Modes Effects Analysis Toolkit

This class outline maps directly to the below process for performing a software FMEA. This class maps to section 2.0 of the IEEE 1633 Recommended Practices for Software Reliability, 2017.



Who should attend: Reliability engineers, systems engineers, software QA, software test engineers, software management, software architects, software requirements engineers, and acquisitions personnel. Class handouts include the Software FMEA toolkit and the "Effective Application of Software Failure Modes Effects Analysis" book.

Class outline
Introduction - statement of goals for class and schedule Real examples of how software FMEAs were used to find serious defects in a cost effective manner.
1. Preparing the SFMEA. a. Identify the scope of the SFMEA - the riskiest and most critical part of the software. b. Identify the people and artifacts needed to do the SFMEA. Identify the viewpoints that are most applicable for the current phase of development and project risks. c. Identify the ground rules for the SFMEA. d. Identify the failure definition and scoring criteria to be used for the SFMEA
2. Brainstorm past failure modes. Employ a defect root cause analysis or software fault tree analysis.
Lunch Break
3. Identify failure modes for the functional SFMEA viewpoint
4. Identify consequences
5. Mitigate
6. Generate the Critical Items List
Class example - The class will see a real example of a functional SFMEA
Identify failure modes for the interface SFMEA viewpoint. Interface FMEAs analyze failure modes between software, firmware and hardware. Class example - The class will see a real example of an interface SFMEA
Identify failure modes for the detailed SFMEA viewpoint. A detailed design FMEA is performed on the design or code.
Identify failure modes for the maintenance SFMEA viewpoint. A maintenance process FMEA analyzes the failure modes related to how people support the software once it is deployed. The focus is on failure modes that would cause previously functional software to stop functioning.
Lunch
How to perform a vulnerability SFMEA. This is a detailed SFMEA that focuses on the design and coding failure modes that are also related to vulnerabilities
How to perform a production process FMEA. A production process FMEA analyzes the failure modes related to how people produce the software product. It's possible for the requirements, design and code to be working, but for the software to be unusable because there is no source control.
How to perform an installation process FMEA. An installation process FMEA analyzes the failure modes related to an end user's or system installation. For example, the software could be working properly but the installation of it might fail. Or the end user may have an incorrect user's manual and be unable to use the software. Class exercise - The entire FMEA process will be executed from analyzing resources to improving the product.
How to NOT do a SFMEA
Closing, Q & A
Optional third day for onsite courses. The third day is spent doing FMEAs on your product and process with the guidance of the instructor.