

Software Design for Reliability (DfR) 2-Day Seminar

Module #	Detailed Module Outlines
1	<p><u>Introductions</u> Introduction of Trainer Agenda Review Training Format and Expectations</p>
Group Exercise #A	<p><u>Group Exercise – Reliability Self-Assessment</u> Objective: Have attendees attempt to provide an assessment of their product software in order to focus on the terminology for expressing software reliability and uncover their underlying confidence in a DfR approach.</p>
2	<p><u>Software Reliability Basic Concepts</u> Software Quality vs. Reliability A Practical Definition Software Reliability Terminology: Defects, Faults and Failures Software Failure Classification Failure Rates, Failure Distributions and MTTF Software Availability and Failure Rates Input Robustness Software Fault Tolerance Summary of Reliable Software Characteristics</p>
3	<p><u>A “Best Practices” Approach to Developing Reliable Software</u> The Paths to Reliable Software Software DfR Options <ul style="list-style-type: none"> • Formal Methods • H/W-Based Approach • Best Practices Approach Software Development Best Practices <ul style="list-style-type: none"> • Comparing Development Behavior Patterns • Best Practices from “Best in Class” Companies • Weakness within Development Organizations The Defect Removal Strategy <ul style="list-style-type: none"> • Defect Phase Propagation • Integrating Reliability Practices into the Development Lifecycle • Comparing Defect Removal Practices • Case Study: Quantifying the Software Reliability Investment DfR Based on a “Best Practices” Approach <ul style="list-style-type: none"> • The Defect Lifecycle and DfR Goals • Summary of DfR Best Practices </p>
4	<p><u>Reliability Measurements, Metrics & Data Analysis</u> Summary of the Reliability Measurements and Metrics to be Tracked Defect Removal Efficiency</p>

Software Design for Reliability (DfR) 2-Day Seminar

Module #	Detailed Module Outlines
	Project Critical Defect and Failure Tracking Failure Rate Data Data Analysis <ul style="list-style-type: none"> • Failure Density Analysis • Failure Causal Analysis
5	<u>Software Reliability Modeling</u> Overview of Predictive Modeling <ul style="list-style-type: none"> • Examples of Tools and Models Overview of Estimation Modeling <ul style="list-style-type: none"> • Examples of Tools and Models
6	<u>Developing a Software Reliability Plan</u> Summary of the Reliability Measurements and Metrics to be Tracked Defect Removal Efficiency Project Critical Defect and Failure Tracking Failure Rate Data Data Analysis <ul style="list-style-type: none"> • Failure Density Analysis • Failure Causal Analysis
7	<u>Unit Test Phase Practices</u> Software Robustness Failure Modes Input Equivalence Classes (ECs) <ul style="list-style-type: none"> • Guidelines for Defining ECs • EC Example • Augmenting ECs: Boundary Value Analysis Coverage-Based Unit Testing <ul style="list-style-type: none"> • Effects of Increased Unit Testing • Types of Code Coverage • Code Coverage Standards Unit Testing Strategies <ul style="list-style-type: none"> • Testing Assessments
8	<u>System Test Phase Practices</u> What is Reliability Testing Reliability Growth Usage Profiles <ul style="list-style-type: none"> • Testing Profile Development • Operational Profiles Generating Software Reliability Estimates
9	<u>Software Fault Tolerance Sample Techniques</u> The Types of Fault Tolerance Requirements for Software Fault Tolerant Techniques

Software Design for Reliability (DfR) 2-Day Seminar

Module #	Detailed Module Outlines
	Sample Software Fault Tolerance Techniques <ul style="list-style-type: none">• Thread/Process Monitoring• Thread/Process Recovery• System Resource Monitoring and Recovery• Software Rejuvenation and Reentrancy
10	<u>Wrap-up</u> Q&A Concluding comments