System Software Reliability

This training course is tailored for reliability engineers, systems engineers, quality assurance engineers, and software engineers and testers. Featuring hands-on software reliability measurement, analyses and design, it is intended for those individuals responsible for measuring, analyzing, designing, automating, implementing or ensuring software reliability for either commercial or government programs. Practical approaches are stressed with many examples included. It is not necessary to have a software background or a reliability background to attend the course, however, either or both are helpful. Hand-outs include a training course manual and a copy of Ann Marie Neufelder's "Ensuring Software Reliability."

Course Instructor

Ann Marie (Leone) Neufelder is the President of SoftRel, a software reliability research, development and testing organization. For more than a decade Ann Marie has been measuring and improving software reliability on a variety of systems in defense and commercial industries. She co-authored two industry standards on software reliability and has published "Ensuring Software Reliability" with Marcel Dekker Inc. Her experience is in the practical hands on application of software reliability measurement and her course reflects this. Ann Marie is a 1983 graduate of Georgia Tech.

Course Contents

Course Introduction

History of Software Reliability

Software Reliability Defined

1. Software Reliability Definition
2. Software Reliability Terms
3. Sources of Software Faults

Software Life Cycle

1. Life Cycle Activities
2. Life Cycle Models
3. Fault, Manpower and Cost Profiles Over Life Cycle

Factors That Impact Software Reliability

1. Application Type
2. Methodologies
3. Product Characteristics
4. Testing/Verification
5. Schedule
6. Maintenance
7. Operational Profile

Overview of Software Reliability Models

1. Types of Software Reliability Models
2. Nomenclature Used in Modeling
3. Assumptions of the Models

Data Required for Models

1. Types of Data
2. Minimum Fault Data Needed
3. Setting Up a Data Collection System
4. Troubleshooting Bad Data

Software Reliability Prediction Models

1. Prediction Models
2. Rome Laboratory TR-92-52
3. Rome Laboratory TR-92-15
4. Musa's Execution Time Model
5. Putnam's Model
6. Historical Data Collection

Software Reliability Estimation Models

1. Objectives
2. Types of Estimation Models
3. Fault Count
    A. Exponential
    B. Shooman Model
    C. Lloyd-Lipow Model
    D. Musa's Basic Model
    E. Musa's Logarithmic Model
    F. Goel-Okumoto Model
    G. Historical Data Collection Model
    H. Weibull Models
4. Test Coverage Models
    A. IEEE Test Coverage Model
    B. Leone's Test Coverage Model
    C. Test Success Model
5. Tagging Models
    A. Seeding
    B. Dual Test Group Model
6. Bayesian Models
7. Thompson and Chelson's Model
8. Goodness of Fit

Software Reliability Metrics

1. Objectives
2. Metrics to Use Based on Your Process Capability
3. Metrics Used in Industry
4. Misusing Metrics

Software Fault Trees

1. Why Fault Trees are Used on Software
2. Applying Fault Trees to Software
3. Software Fault Tree Analysis Example

Software FMEAs

1. Why FMEAs are Used on Software
2. Applying FMEAs to Software
3. Example of Software FMEA

System Reliability Software Redundancy

1. Series Configuration
2. Mission Oriented
3. Semi-Markov
4. Parallel Concurrent
5. Voting Redundancy

Improving Software Reliability

1. Evaluating Your Own Product and Process
2. Techniques for Improving Software Reliability

Managing Software Reliability

1. Matrix of Responsibilities
2. Cost Benefit of Improvement