

# Verification, Validation, and Test Technology

## Prospective Course Outline

Proposed Class Schedule, Software and System V&V

<b>Objectives of V&amp;V</b>	* VV&T to prevent deterministic failures	* Process techniques	* Markov models	* Application: DO-178
Characterization of Failures		Analysis techniques (overview)	Collecting and evaluating data	
Discussion of specific failures	Process causes of random failures	Fault tree and event tree analysis	Looking for precursors of failures	Summary & Recommendations
Software failures – Random vs. deterministic	Prevention and mitigation of random failures	<b>Failure modes &amp; effects analysis as a process tool</b>	<b>Reliability growth models in process evaluation</b>	

\*Plus 20 minute review of the previous day's classes

## Objectives and Format

“One of the things we kept in mind during the course of our review is that in the conduct of space missions, you get only one strike, not three” – Tom Young, *MPIAT Report*

1. Special requirements for FAA VV&T
  - 1.1 Typical commercial VV&T : Has this been done right?
  - 1.2 FAA: Have all possible sources of failures been identified, addressed, corrected, verified?
2. Guides for failure prevention
  - 2.1 Preliminary Hazard Analysis (PHA) – NHB 1700.1 (V1-B)
  - 2.2 Software Safety – NASA STD-8719.13A
  - 2.3 Guidebook for Safety Critical Software – NASA-GB-1740.13-96
3. Format of course
  - 3.1 Understanding causes of failures
  - 3.2 Classification of failures
  - 3.3 Role of VV&T
  - 3.4 Prevention/mitigation techniques
  - 3.5 Lessons learned: data collection and evaluation
  - 3.6 Useful statistics and their limitations
  - 3.7 Applying it to the FAA missions

## Characterization of Failures

1. By life cycle phase
2. By medium

- 2.1 Hardware – types of hardware failures
- 2.2 Firmware \*
- 2.3 Software \*
- 2.4 Skinware - limitations of human capabilities must be recognized
  - \* detailed discussion deferred to next session

- 3. Severity / frequency relation
- 4. Complexity of conditions that produce a failure as an indication of maturity
- 5. By review opportunity
  - 5.1 Design review
  - 5.2 Formal verification
  - 5.3 VV&T team
  - 5.4 Subsystem test
  - 5.5 Integration test
  - 5.6 Mission simulation

#### Discussion of specific failures

- 1. Software test failures – role of multiple exception conditions
- 2. Multi-version software test – multiple exceptions as a screening tool
- 3. Mars Polar Lander failure – discussion of each “Flag” in the JPL report
- 4. Ariane 5 failure – hazards in adopting legacy software, faulty mitigation logic

#### Software failures – random vs. deterministic

- 1. Random = detailed cause not established
  - 1.1 Maybe hardware failure
  - 1.2 Non-repeatable software failure – multiple exception conditions
  - 1.3 Hardware – software interactions
  - 1.4 Insufficient investigation
- 2. Deterministic = reproducible

- 2.1 Requirement not recognized or implemented
- 2.2 Hardware – software interactions
- 2.3 Run-time environment interactions
- 2.4 Configuration management

### VV&T to prevent deterministic failures

- 1. Formal Methods as a means of preventing software failures
  - 1.1 Requires a stated requirement
  - 1.2 Most effective when used by domain expert rather than FM specialist
  - 1.3 Has been found effective for communication and security protocols
- 2. Assessment of the development process – who, when, where (including subcontractors)
  - 2.1 Requirements
  - 2.2 Design
  - 2.3 Implementation
  - 2.4 Test
- 3. Failure Reporting and Corrective Action System
  - 3.1 Responsibility for (a) system operation (b) failure review
  - 3.2 Wide participation
  - 3.3 Failure classification – local vs. Center vs. NASA wide
  - 3.4 Awareness of other failure reporting agencies
  - 3.5 Root cause evaluation
- 4. Use of modeling techniques
  - 4.1 Fault density models
  - 4.2 Software reliability models

## Causes of random failures

1. Unknown environments
  - 1.1 System interactions
  - 1.2 Launch environment
  - 1.3 Space environment
2. Noise effects
  - 2.1 Noise susceptibility of semiconductors
  - 2.2 Noise susceptibility of sensors
  - 2.3 Noise in communications
3. Load-Strength relations
  - 3.1 Mechanical
  - 3.2 Electrical
  - 3.3 Logical (speculative)

## Prevention and mitigation of random failures

1. Redundancy
  - 1.1 Same element redundancy – effective against independent failures
  - 1.2 Diverse element redundancy – effective against some non-independent failures
  - 1.3 Analytical redundancy – limited possibilities, but cost-effective
  - 1.4 Time redundancy
  - 1.5 Communication redundancy
2. Margins
  - 2.1 Noise margins
  - 2.2 Mechanical and electrical design margins

### 2.3 Timing margins

## 3. VV&T responsibilities

- 3.1 Assess independence of failure mechanisms
- 3.2 Assess independence of actuation mechanisms
- 3.3 Assess fault detection and recovery coverage

## Process Techniques

### 1. For all reviews

- 1.1 Responsibility for conduct
- 1.2 Participation by project management, mission specialists, developers, testers. Contractors
- 1.3 Methodology of review
- 1.4 Utilization of PHA
- 1.5 Adequacy of test requirements and facilities
- 1.6 Implementation of findings
- 1.7 Action on changes after review

### 2. Requirements review

- 2.1 Validation of legacy components
- 2.2 Requirements for multiple exception handling
- 2.3 Statement of acceptable risk

### 3. Design review

- 3.1 Extent of innovation in design techniques
- 3.2 Validation of legacy components
- 3.3 Use of new components
- 3.4 Fall-back alternatives

4. Implementation review
  - 4.1 Review of recent change activity
  - 4.2 Evidence of maturity
  - 4.3 Thoroughness of unit test
  
5. Test Review
  - 5.1 Adherence to test plan
  - 5.2 Documentation of test plan changes
  - 5.3 Documentation of requirements and design changes after formulation of test plan
  - 5.4 Review of test results by mission specialist and design team

#### Analysis techniques – overview

1. Independent analysis as part of VV&T
  - 1.1 Means of gaining insight into mission and design
  - 1.2 Validation of analyses of development team
  - 1.3 Validation of currency of requirements
  
2. Types of analyses
  - 2.1 Fault tree
  - 2.2 Event tree
  - 2.3 Failure Modes and Effects
  - 2.4 Timing
  - 2.5 Coverage of redundancy provisions

#### Fault tree and event tree analyses

1. Common objectives

- 1.1 Identify mechanisms that can give rise to mission failure
- 1.2 Establish currency of PHA
  
2. Fault tree analysis
  - 2.1 Manual methodology
  - 2.2 Computer based tools
  - 2.3 Interpretation of results
  - 2.4 Interaction with developers
  
3. Event tree analysis
  - 3.1 Purpose
  - 3.2 Difference from FTA
  - 3.3 Manual methodology
  - 3.4 Automated methods
  - 3.5 Interpretation of results

## Failure Modes and Effects Analysis

1. Bottom-up technique
  - 1.1 Historically a parts-level technique
  - 1.2 Limitations when used with ICs and software
  - 1.3 Effective in combination with FTA
  
2. Methodology
  - 2.1 Introduction to format and MIL-STD-1629
  - 2.2 Grouping of components by local failure effects
  - 2.3 Classification of “next higher effects”
  - 2.4 Classification of “system level” effects
  - 2.5 Interpretation of results

## Markov Models

1. Introduction to system states
  - 1.1 Viewing operation of a system as state transitions
  - 1.2 System failures as state transitions
  - 1.3 Difference between state transitions and reliability block diagrams
  - 1.4 Solution as a system of differential equations
  - 1.5 Computer based solutions
2. Example of a redundant system with fractional coverage
  - 2.1 Defining the states
  - 2.2 Defining arcs
  - 2.3 Manual solution
  - 2.4 Significance of results
  - 2.5 Sensitivity studies
3. Example of a repairable system
  - 3.1 Defining the arcs
  - 3.2 Manual solution
  - 3.3 Significance of results
  - 3.4 Sensitivity studies

## Collecting and evaluating data

1. Data Types to be collected for VV&T
  - 1.1 Design margins (noise, strength, timing)
  - 1.2 Failure reports
  - 1.3 Failure and rejection statistics

- 1.4 Test problem reports
- 2. Evaluation
  - 2.1 Quality (prepared and reviewed by responsible personnel, completeness)
  - 2.2 Consistency (variations between early and late data, between subsystems)
  - 2.3 Evaluation by development team
  - 2.4 Independent VV&T evaluation
- 3. Recommendations
  - 3.1 For data collection
  - 3.2 For data evaluation

#### Looking for precursors of failures

- 1. Individual failures of redundant channels
  - 1.1 Adequacy of reporting
  - 1.2 Failure frequency – comparison with predicted
  - 1.3 Failure analysis and corrective measures
  - 1.4 Applicability to similar equipment or environment
- 2. Evidence of inadequate margins
  - 2.1 Are critical parameters recorded in numerical format (as opposed to pass/fail)
  - 2.2 Analysis performed on mean and standard deviation of results
  - 2.3 Trend analyses
  - 2.4 Are original requirements still valid?
- 3. Cluster analysis
  - 3.1 Evidence of clusters in time
  - 3.2 Evidence of clusters by component type

### 3.3 Evidence of clusters by environment

#### Reliability growth models and related statistics

1. Applicability
  - 1.1 Developed software
  - 1.2 New hardware designs
  - 1.3 New interfaces
2. Assumptions
  - 2.1 Failure rate is proportional to residual faults
  - 2.2 Faults are uncovered at a uniform rate
  - 2.3 Faults are removed once uncovered
3. Model validation
  - 3.1 Graphical failure rate presentations
  - 3.2 Analytical verification of decreasing failure rate
  - 3.3 Evaluation against requirements
4. Limitations of statistical methods
  - 4.1 The past is not necessarily prelude to the future
  - 4.2 Don't take a trend out the window
  - 4.3 Not reliable for single string threads

#### Mission to Venus

1. Form development teams
2. Establish requirements

3. Exchange with other teams for VV&T

4 Critique of VV&T

### Summary

1. What are key elements of VV&T
2. What limitations must be accepted?
3. What methods do you rate as most effective?
4. What are your recommendations for improved VV&T processes?

### Opening remarks

1. Root causes to be addressed by VV&T
  - Software complexity
  - Hardware complexity and noise susceptibility
  - Unknown environments
2. Defenses
  - Redundancy and back-up provisions
  - Adequate margins (timing, strength, noise)
  - Utilization of all data that may impair mission

