



ILLINOIS INSTITUTE
OF TECHNOLOGY

Transforming Lives. Inventing the Future.

www.iit.edu

SOFTWARE QUALITY MANAGEMENT

CSP587

Prof. Dennis Hood
Computer Science

Reading

- Verification and Validation
- Reading
 - Ch. 7 – Verification and Validation
 - Case Study: [Ariane 5 failure](#)
- Objectives
 - Explore the testing process
 - Discuss different types of “testing” conducted throughout the life cycle including evaluations, reviews, assessments, verifications, audits, etc.
 - Recognize the (near) impossibility of testing everything and develop strategies for maximizing effectiveness
 - Appreciate the need for thorough testing – even when reusing trusted components

Topics for Discussion

- State the objective of testing in measurable terms.
- Explain why it is likely not feasible to prove that a software system is perfect.
- Discuss the potential for risk-based testing to increase both effectiveness and efficiency.
- Testing for ease-of-use requires different strategies than “traditional” (functional) testing. Explain why and discuss a strategy.
- Discuss the potential for automated test tools to provide greater test coverage.
- What does the Ariane 5 failure tell us about reuse?



Week 8

Verification and Validation

The Nature of Accidents

- Previously – Mechanical Failure
 - Accidents often occur due to fatigue and misuse
- Modern – System Crashes
 - Accidents often occur at points of interaction between components / partners (including human users)
- Testing must explore these vulnerabilities
 - Confirmation (with objective evidence) of requirements fulfillment
 - Verify that the requirements are addressed
 - Validate that the system performs correctly

Case Study: Ariane 5 Launch Failure

- Reuse plan
 - Inertial reference software performed successfully in the Ariane 4, so it was reused in the Ariane 5
 - The code contained “extras”
 - The Ariane 5 had more powerful engines and therefore could achieve greater velocities
- Failure
 - Code (*that was not required*) attempted to convert a fixed-point number to an integer
 - Ariane 4 had never generated such a large number
 - The code generated an exception and shut down the navigation system (both primary and backup)
 - The code that failed was not tested since it wasn't actually used for Ariane 5 (but was left to execute anyway)

Checks and Balances

- Work products should be reviewed to discover defects
 - Work products come in many forms
 - So do defects
 - Undiscovered defects will be passed downstream
- Defect discovery is much more effective when the reviewer is independent of the production team
 - Why?
 - How do we achieve necessary independence while maintaining sufficient knowledge?

Software Testing

- Traditionally applied late in the SDLC
- Common sense has moved it up
- Testing differs from reviews in that it involves exercising work products
 - Specialized testing team
 - Approved test procedures
 - Approved test cases
- Objectives
 - Direct
 - Discover defects and achieve acceptability
 - Indirect
 - Collect data

Strategies

- Big-bang vs. incremental
 - More comprehensive and final testing of the full package, vs.
 - Sooner, quicker yet less complete testing of modules
- Top-down vs. bottom-up
 - Start at the “main” module and drive down into the “functionality”, vs.
 - Focusing on the details first

Classifications

- Black-box
 - Focus on output
 - Equivalence classes (sets of inputs producing the same output)
 - Boundary values
 - Documentation testing
 - Availability and reliability
 - Stress
 - Security
 - Usability
- White-box
 - Focus on paths through the code
 - Correctness of calculations
 - Path and line coverage
 - Cyclomatic complexity metrics
 - Qualification and reusability

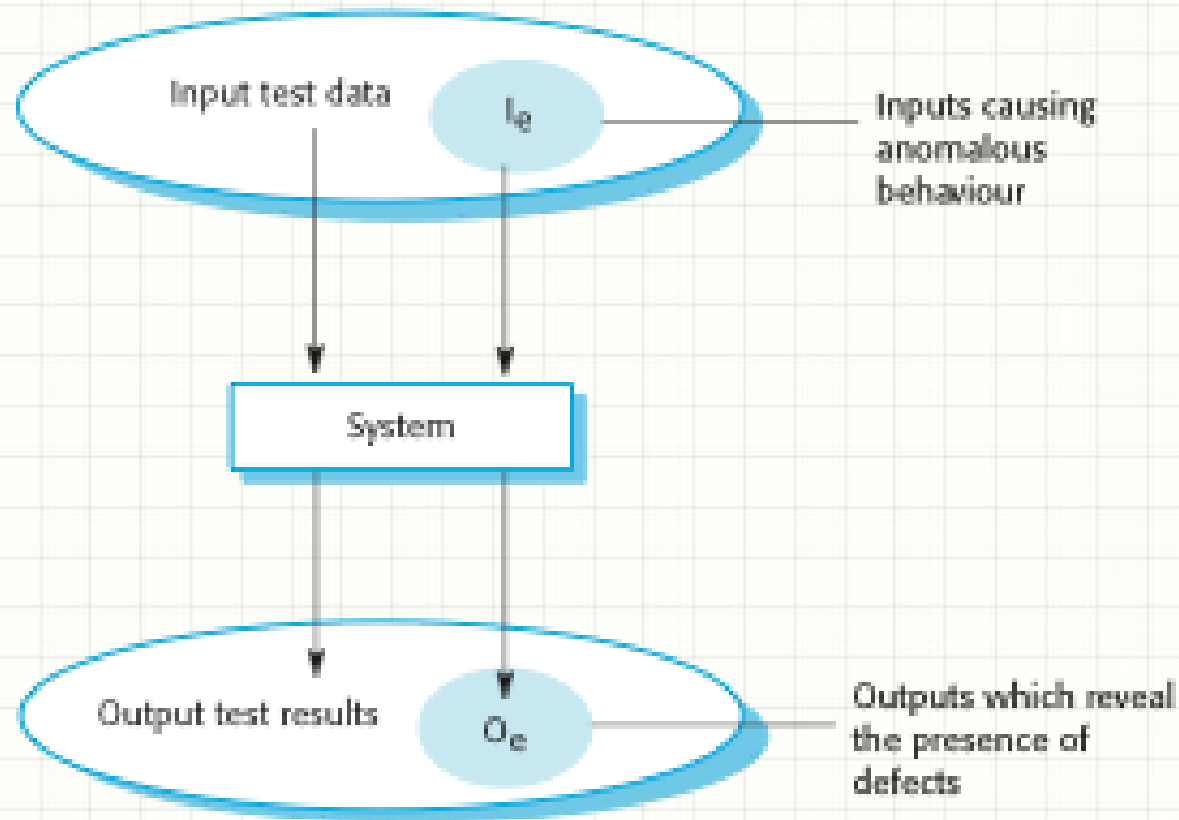
Implementation

- The testing process
 - Determine the test methodology
 - Plan the tests
 - Design the tests
 - Test away!
- Test cases
- Automation
- Alpha and beta testing

The Role of Testing

- Goals
 - Demonstrate correctness, completeness, etc.
 - At least one test per requirement
 - Tests to fully exercise features
 - Discover any defects
 - Or at least gain confidence that all have been discovered
- V&V
 - Validate that we are building the right product
 - Verify that we are building it right

Testing Model



Fit for Purpose

- The system is good enough for its intended use
- Software purpose – the more critical the system, the more important that it is reliable
- User expectations
 - Tolerance for defects
 - Sense of value in the system's capabilities
- Marketing environment
 - Competition can both drive organization's to strive for greater quality OR motivate them to release systems without fully testing
 - User expectations of quality are related to price

Development Testing

- Testing done local to the development effort
 - Often done informally by the developer
 - Primarily for defect detection – “debugging”
- Levels of granularity
 - Unit testing – exercise the functionality of logical units of the system
 - Component testing – verify proper operation of interacting entities such as objects
 - System testing – exercise the system as a whole for proper operation, exception handling, tolerance of load, etc.

Test-driven Development

- Interleave testing and code development
 - Develop a portion of code and its associated test(s)
 - Move onto the next increment of code only when current increment passes testing
- TDD approach
 - Forces partitioning of system into portions
 - Ensures “clean” code (in portions)
 - Utilize automated testing
 - Facilitates deeper understanding of the system
 - Provides a level of documentation
 - Needs to be supported by occasional “big picture” assessment

Release Testing

- Establish a “fit for use” version of the system
 - Requires independent verification
 - Focus is on validation more than defect discovery
 - Usually “black-box” in nature (ins and outs)
- Requirements testing
 - Requirements should be testable
 - Demonstrate proper implementation of system requirements
- Scenario testing
 - Verify “realistic” operation
- Performance or load testing

User Testing

- User acceptance is the ultimate goal of systems development
- User involvement is critical to successful development
- Users should be involved in test planning and execution
- Approaches
 - Alpha – within development process
 - Beta – “field” testing of a preliminary release
 - Acceptance – users determine “fitness”

Test Cases

- Effectiveness
 - Efficiently discover defects
 - Credibly show proper operation
- Efficiency
 - Repeatable
 - Self-documenting
 - Easy to develop and maintain
- Strategies
 - Test normal AND abnormal
 - Use realistic data
 - Test boundaries

Interface Testing

- Exercise the interface
 - Parameter passing
 - Return values and types
 - Synchronization
 - State management
- Box testing
 - Predictable output for given input
 - “Correctly incorrect” output for given improper input

Test Planning

- Testing process description
- Requirements traceability
- Items to be tested
- Schedule
- Results recording procedures
- Required hardware and software
- Constraints