

Managing the Test Process



**Idaho State
University**

Computer
Science

Isaac Griffith

CS 4422 and CS 5599
Department of Computer Science
Idaho State University

ROAR

Outcomes

At the end of Today's Lecture you will be able to:

- Understand how testing integrates throughout the entire software development lifecycle



Overview

- Organizations that wait until the end of development to test compromise the testing process.
- Testers cannot show up at the last minute and make a bad product good, quality must be part of the process from the start
- Thus testing must be integrated at each stage of development
 - By doing so we can make dramatic improvements in the effectiveness and efficiency of testing in order to positively impact overall software quality.



Software Process

- Though there are many different software processes found in SE literature the following stages are common to most (if not all)
 - Requirements analysis and specification
 - System and software design
 - Intermediate design
 - Detailed design
 - Implementation
 - Integration
 - System development
 - Operation and maintenance



Requirements

Objectives

- Ensure requirements are testable
- Ensure requirements are correct
- Ensure requirements are complete
- Influence the software architecture

Activities

- Setup testing requirements
 - choose testing criteria
 - obtain or build support software
 - define testing plans at each level
 - build test prototypes
- Clarify requirement items and test criteria
- Develop project test plan



Design

Objectives

- Verify mapping between requirements specification and system design
- Ensure traceability and testability
- Influence interface design

Activities

- Validate design and interface
- Design system tests
- Develop coverage criteria
- Design acceptance test plan
- Design usability test (if necessary)

Intermediate Design

- **Test Influence Goal** - to influence detailed design

Objectives

- Avoid mismatches of interfaces
- Prepare for unit testing

Activities

- Specify system test cases
- Develop integration and unit test plans
- Build or collect test support tools
- Suggest ordering class integration

Detailed Design

- **Test Influence Goal** - influence the implementation and unit and integration testing.

Objectives

- Be ready to test when modules are ready

Activities

- Create test cases (if unit)
- Build test specifications (if integration)

Implementation

- **Test Influence Goal** - efficient unit testing ensures early integration and system testing.

Objectives

- Efficient unit testing
- Automatic test data generation

Activities

- Create test case values
- Conduct unit testing
- Report problems properly



Integration

- Integration and integration testing begin as soon as needed components pass unit testing

Objectives

- Efficient integration testing

Activities

- Perform integration testing



System Development

- **System Testing** - compares the software system to its original objectives (i.e. validating that it meets functional and non-functional requirements)
- **Acceptance Testing** - ensures the completed system meets the customer's needs (thus requires users) - should be started as soon as system testing completes
- **Usability Testing** - evaluates the UI of the software (thus requires users)

Objectives

- Efficient system testing
- Efficient acceptance testing
- Efficient usability testing

Activities

- Perform system testing
- Perform acceptance testing
- Perform usability testing



Operation/Maintenance

- After deployment uses will
 - Find new problems
 - Request new features
- After a change the system should be regression tested
 - Ensures updated software maintains its functionality
 - Tests for new/modified functionality

Test Process Implementation

- Quality in the dev process requires professional ethics
- Both devs and testers must choose to **put quality first**
- Start testing early
- Fight the desire to take shortcuts that reduce quality
 - Stand up for what is right
 - Document your objections
 - Don't be afraid to walk away
- Manage test artifacts
 - test designs docs, tests, test results and automated support should be version controlled
 - track test criteria-based source of tests



Are there any questions?