SM-515
Software Verification & Validation

Fall 2002 Semester

Lecture 10: Software Testing II
16/12/2002

Dr. Kivanç DINÇER

TÜBITAK-UEKAE/ILTAREN
http://www.uao.tubitak.gov.tr/

Testing

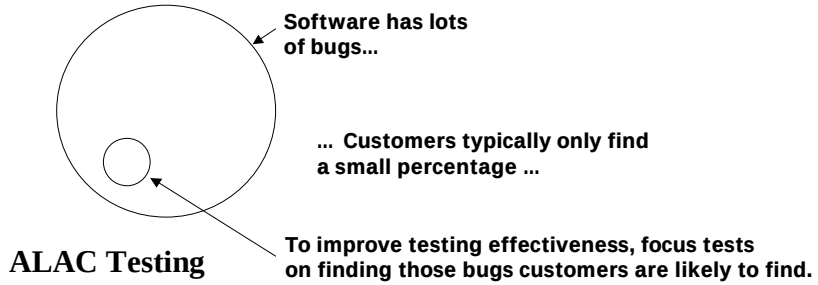
- Testing is the process of executing programs with the intention of finding errors.
- Testing can show the presence of bugs but never their absence.

(Common misconceptions about testing, difficulty of testing)

Test Methods

- White box or glass box testing
- Functional or black box testing
- Top-down and bottom-up testing
- Act-like-a-customer (ALAC) testing
  - Tests are developed based on knowledge of how customers use your software.

ALAC Testing



Testing Levels and Test Medhods

Test Level	Objectives	Performed by	Test Environment	Test Methods
Unit	Find bugs in logic, data, and algorithms in individual modules	Software engineers	Isolated. Stubs and scaffolding may be required	White box
Integration	Find bugs in interfaces between modules	Software Engineers	Isolated or simulated. Stubs and scaffolding may be required.	White box Top-down and Bottom-up
Validation	Determine if software meets SRS	QA	Actual	Functional and ALAC
Regression	Determine if software still meets SRS in light of changes	QA	Actual	Functional and ALAC
Acceptance	Determine if software meets customer requirements	Customer, QA, or project team	Actual (usually at customer site)	Functional and ALAC

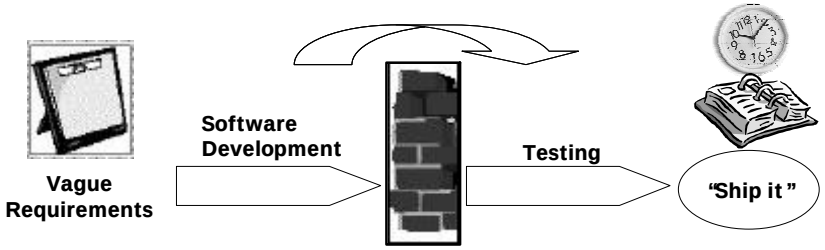
Test Types

- Functional
- Algorithmic
- Positive
- Negative
- Safety-related
- Compatibility
- Life
- Usability
- Boundary
- Startup and shutdown
- Configuration
- Platform
- Load / Stress
- Security
- Performance
- Documentation
- Timing
- Error checking
- Power failure
- Out of resources / space
- Installation
- Upgrade
- Volume scalability
- Throughput/performance

When planning a testing effort, choose an appropriate mix of test types to increase the likelihood that defects will be uncovered.

Concurrent Development/Validation Testing Model

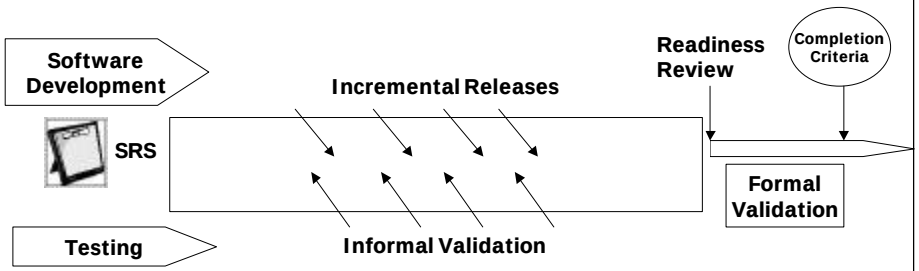
- In organizations where the relationship between development and testing is not understood, validation testing often happens as illustrated below.



Typical validation testing process

Concurrent Development/Validation - 2

- To increase the ROI from your testing effort, you need a good understanding of testing levels, methods and types.
  - Ideally, you should plan to perform a mix of testing activities, commensurate with risk and business objectives.



Concurrent development/validation testing model

Informal Validation

- As development continues, incremental releases are provided to QA, which develops tests in the same order as the incremental features are being developed.
  - By the time coding is completed, all validation tests have been written and run at least once. Thus majority of problems should have been found, reported and corrected.
- This activity is called informal validation, because tests are run informally.
  - Provides an opportunity for validation tests to be developed and debugged early in the software development process
  - Provides early feedback to software engineers
  - Results in formal validation being less eventful, since most of the problems have already been found and fixed.

Validation Readiness Review

- The purpose of this review is to ensure that everything is in place before beginning formal validation.
  - Starting formal evaluation prematurely results in wasted effort, increased frustration, and pressure to release products with far too many defects.

Differences Between Informal and Formal Validation

- Informal validation:
  - Developers can make any changes needed in order to comply with the SRS
  - QA runs tests and makes changes necessary in order for the tests to comply with the SRS
- Formal validation
  - Developers can only fix bugs reported during formal validation testing. No new features can be added.
  - QA runs the same set of tests run during informal validation. No new tests can be added.

Formal Validation / Entry Criteria

- The test plan should define the criteria that should be met before formal validation can begin:
  - Software development has been completed
  - The test plan has been reviewed, approved, and is under document control
  - A requirements inspection has been performed on the SRS
  - Design inspections have been performed on the SDDs
  - Code inspections have been performed on all critical modules
  - All test scripts have been completed and software validation test procedure document has been reviewed, approved, and placed under document control.
  - Selected test scripts have been reviewed.
  - All test scripts have been executed at least once
  - CM tools are in place and all source code is under configuration control
  - Software problem reporting procedures are in place
  - Validation testing completion criteria have been developed, reviewed, and approved.

Formal Validation / Entry Criteria

- The test plan should define the criteria that should be met before formal validation can begin:
  - Software development has been completed
  - The test plan has been reviewed, approved, and is under document control
  - A requirements inspection has been performed on the SRS
  - Design inspections have been performed on the SDDs
  - Code inspections have been performed on all critical modules
  - All test scripts have been completed and software validation test procedure document has been reviewed, approved, and placed under document control.
  - Selected test scripts have been reviewed.
  - All test scripts have been executed at least once
  - CM tools are in place and all source code is under configuration control
  - Software problem reporting procedures are in place
  - Validation testing completion criteria have been developed, reviewed, and approved.

Formal Validation

- At this point, software changes are restricted to changes required to fix bugs. No new functionality can be added.
- Activities:
  - The same tests that were run during informal validation are re-executed again and results recorded.
  - Software Problem Reports (SPRs) are submitted for each test that fails.
  - SPR tracking is performed and includes the status of all SPRs (i.e., open, fixed, verified, deferred, not a bug)
  - For each bug that is fixed, the SPR identifies the modules that were changed to fix the bug.
  - Baseline change assessment is used to ensure that only those modules that should have been changed actually have chagend and that no new features have slipped in.
  - Informal code reviews are selectively conducted on changes modules to ensure that new bugs are not being introduced.

Formal Validation

- At this point, software changes are restricted to changes required to fix bugs. No new functionality can be added.
- Activities:
  - The same tests that were run during informal validation are re-executed again and results recorded.
  - Software Problem Reports (SPRs) are submitted for each test that fails.
  - SPR tracking is performed and includes the status of all SPRs (i.e., open, fixed, verified, deferred, not a bug)
  - For each bug that is fixed, the SPR identifies the modules that were changed to fix the bug.
  - Baseline change assessment is used to ensure that only those modules that should have been changed actually have chagend and that no new features have slipped in.
  - Informal code reviews are selectively conducted on changes modules to ensure that new bugs are not being introduced.

- Activities:
  - Time required to find and fix bugs (find-fix cycle time) is tracked.
  - Regression testing is performed
  - Track test status (i.e., passed, failed, or not run)
  - Record cumulative test time for software reliability growth tracking.

When to stop testing?

- It is very important to have objective, measurable completion criteria defined, reviewed. and approved early in the development process.
  - All test scripts have been executed
  - All SPRs have been satisfactorily resolved
  - All changes made as a result of SPRs have been tested
  - All related documentation have been updated to reflect changes during validation testing.
  - The test report has been reviewed and approved.