Testing Process

Testing

- Testing only reveals the presence of defects
- Does not identify nature and location of defects
- Identifying & removing the defect => role of debugging and rework
- Preparing test cases, performing testing, defects identification & removal all consume effort
- Overall testing becomes very expensive : 30-50% development cost

Testing…

- Multiple levels of testing are done in a project
- At each level, for each SUT, test cases have to be designed and then executed
- Overall, testing is very complex in a project and has to be done well
- Testing process at a high level has: test planning, test case design, and test execution

Test Plan

- Testing usually starts with test plan and ends with acceptance testing
- Test plan is a general document that defines the scope and approach for testing for the whole project
- Inputs are SRS, project plan, design
- Test plan identifies what levels of testing will be done, what units will be tested, etc in the project

Test Plan…

- Test plan usually contains
  - Test unit specs: what units need to be tested separately
  - Features to be tested: these may include functionality, performance, usability,…
  - Approach: criteria to be used, when to stop, how to evaluate, etc
  - Test deliverables
  - Schedule and task allocation

Test case Design

- Test plan focuses on testing a project; does not focus on details of testing a SUT
- Test case design has to be done separately for each SUT
- Based on the plan (approach, features,..) test cases are determined for a unit
- Expected outcome also needs to be specified for each test case

Test case design…

- Together the set of test cases should detect most of the defects
- Would like the set of test cases to detect any defects, if it exists
- Would also like set of test cases to be small - each test case consumes effort
- Determining a reasonable set of test case is the most challenging task of testing

Test case design

- The effectiveness and cost of testing depends on the set of test cases
- Q: How to determine if a set of test cases is good? I.e. the set will detect most of the defects, and a smaller set cannot catch these defects
- No easy way to determine goodness; usually the set of test cases is reviewed by experts
- This requires test cases be specified before testing – a key reason for having test case specs
- Test case specs are essentially a table

Test case specifications

Seq.No	Condition to be tested	Test Data	Expected result	successful

Test case specifications…

- So for each testing, test case specs are developed, reviewed, and executed
- Preparing test case specifications is challenging and time consuming
  - Test case criteria can be used
  - Special cases and scenarios may be used
- Once specified, the execution and checking of outputs may be automated through scripts
  - Desired if repeated testing is needed
  - Regularly done in large projects

Test case execution

- Executing test cases may require drivers or stubs to be written; some tests can be auto, others manual
  - A separate test procedure document may be prepared
- Test summary report is often an output – gives a summary of test cases executed, effort, defects found, etc
- Monitoring of testing effort is important to ensure that sufficient time is spent
- Computer time also is an indicator of how testing is proceeding

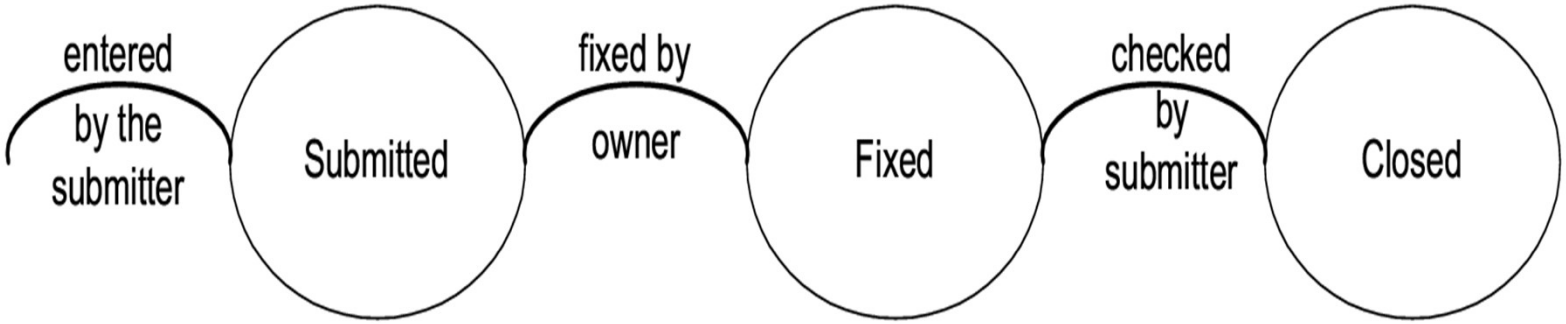
Defect logging and tracking

- A large software may have thousands of defects, found by many different people
- Often person who fixes (usually the coder) is different from who finds
- Due to large scope, reporting and fixing of defects cannot be done informally
- Defects found are usually logged in a defect tracking system and then tracked to closure
- Defect logging and tracking is one of the best practices in industry

Defect logging…

- A defect in a software project has a life cycle of its own, like
  - Found by someone, sometime and logged along with info about it (submitted)
  - Job of fixing is assigned; person debugs and then fixes (fixed)
  - The manager or the submitter verifies that the defect is indeed fixed (closed)
- More elaborate life cycles possible

Defect logging…



Defect logging…

- During the life cycle, info about defect is logged at diff stages to help debug as well as analysis
- Defects generally categorized into a few types, and type of defects is recorded
  - ODC (orthogonal defect classification) is one classification
  - Some std categories: Logic, standards, UI, interface, performance, documentation,..

Defect logging…

- Severity of defects in terms of its impact on sw is also recorded
- Severity useful for prioritization of fixing
- One categorization
  - Critical: Show stopper
  - Major: Has a large impact
  - Minor: An isolated defect
  - Cosmetic: No impact on functionality

Defect logging…

- Ideally, all defects should be closed
- Sometimes, organizations release software with known defects (hopefully of lower severity only)
- Organizations have standards for when a product may be released
- Defect log may be used to track the trend of how defect arrival and fixing is happening