Software testing process

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 specifications

- Test plan focuses on approach; does not deal with details of testing a unit
- Test case specification has to be done separately for each unit
- 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 specifications…

- 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 specifications…

- 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 and analysis

- 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

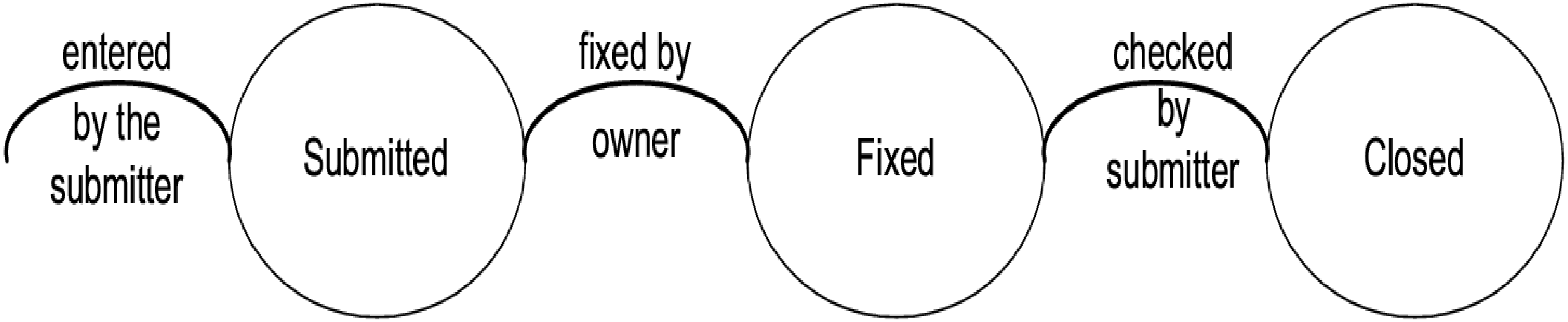
Defect logging and tracking

- 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
  - Orthogonal Defect Classification (ODC) is one classification
  - Some std categories: Logic, standards, UI, interface, performance, documentation,..

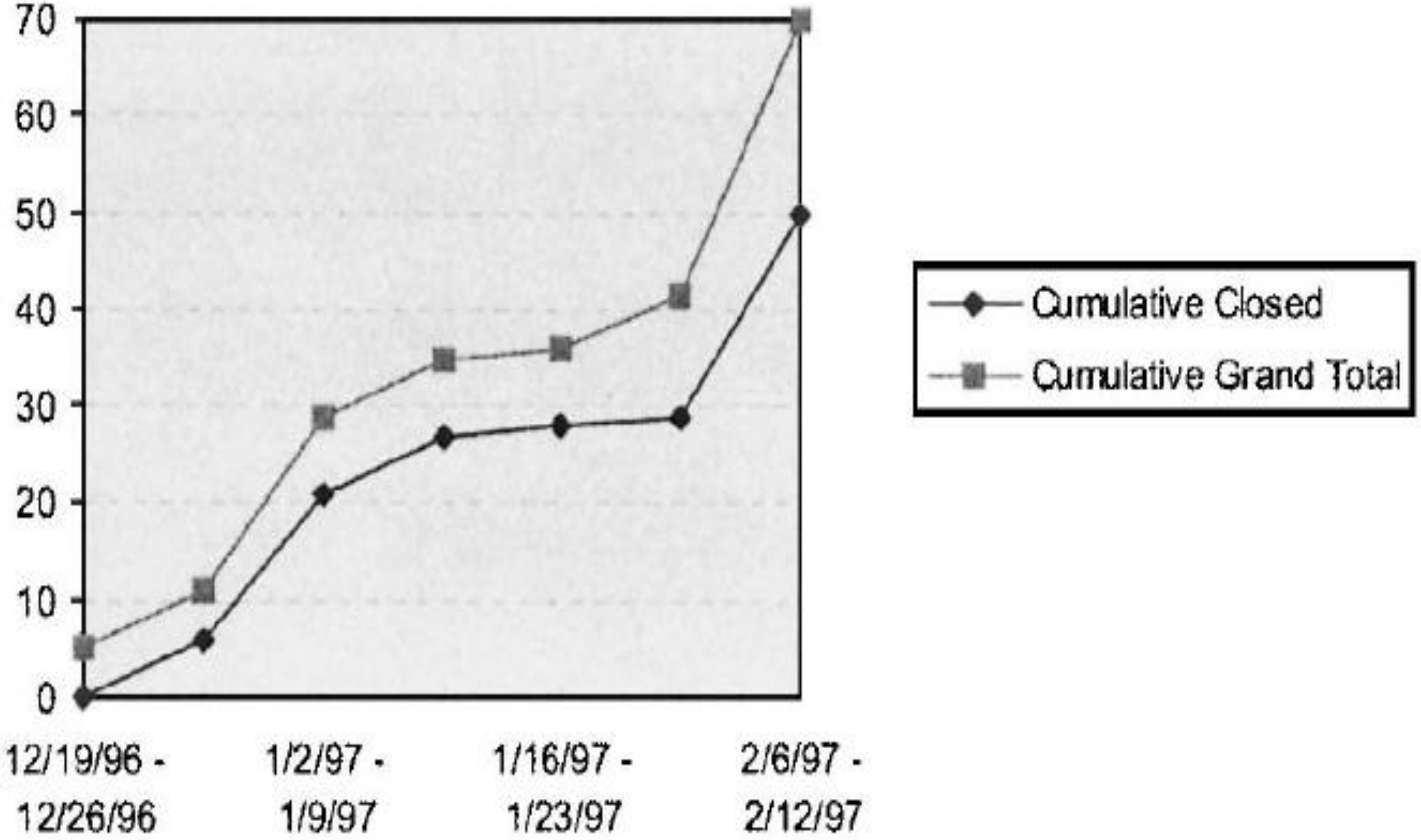
Defect logging…

- Severity of defects in terms of its impact on sw is also recorded
- Severity useful for prioritization of fixing
- One categorization
- **Critical:** Show stopper; affects a lot of users; can delay project.
- **Major**: Has a large impact; Has a large impact but workaround exists; considerable amount of work needed to fix it, though schedule impact is less.

- **Minor**: An isolated defect that manifests rarely and with little impact.
- **Cosmetic**: No impact on functionality; Small mistakes that don't impact the correct working.

Defect logging and tracking…

- 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


70
60
50
40
30
20
10
0
12/19/96 - 12/26/96
1/2/97 - 1/9/97
1/16/97 - 1/23/97
2/6/97 - 2/12/97
Cumulative Closed
Cumulative Grand Total

Summary

- Testing plays a critical role in removing defects, and in generating confidence
- Testing should be such that it catches most defects present, i.e. a high DRE
- Multiple levels of testing needed for this
- Incremental testing also helps
- At each testing, test cases should be specified, reviewed, and then executed

Summary …

- Deciding test cases during planning is the most important aspect of testing
- Two approaches – black box and white box
- Black box testing - test cases derived from specifications.
  - Equivalence class partitioning, boundary value, cause effect graphing, error guessing
- White box - aim is to cover code structures
  - statement coverage, branch coverage

Summary…

- In a project both used at lower levels
  - Test cases initially driven by functional
  - Coverage measured, test cases enhanced using coverage data
- At higher levels, mostly functional testing done; coverage monitored to evaluate the quality of testing
- Defect data is logged, and defects are tracked to closure
- The defect data can be used to estimate reliability, DRE