

COMP354

Software Engineering

Lecture 5

Overview of Testing for Project Phase 1

Testing

Testing is verifying the quality of correctness

Aim of testing: *To discover the presence of errors!!!*

Stages of Testing

Unit Testing testing an individual unit or basic component of the system

eg testing a function SQRT

Module Testing testing a module consisting of several units, to check that their interaction and interfaces are ok

eg testing that pop and push in a stack module provide a last-in-first-out behaviour

Subsystem Testing testing a subsystem consisting of several modules, to check module interaction and module interfaces are ok

eg an output subsystem writing to a database, and keeping an audit trail, and providing rollback services

Integration Testing testing the entire system once all the subsystems have been integrated

Top-down vs Bottom-up Testing

Top-down testing: test subsystems, then modules, then units

need a STUB (prototype) for each non-completed module or unit

a stub simulates the behaviour of the module/unit

may do nothing; may always exhibit the same default behaviour; may only handle those cases needed for tests

Advantages:

catches high-level design errors early: saves cost

Disadvantages:

not so easy to devise tests at this level; additional cost of implementing stubs

(high-level design errors can be caught in design walk-throughs)

Bottom-up testing: test units, then modules, then subsystems

need a DRIVER to supply necessary input, output, data structures to test the unit, module, subsystem

Advantages:

every component is tested before integration into a larger component; makes location of errors easier, since we first assume it is in the integration/interaction of subcomponents rather than within the subcomponents

Disadvantages:

delays detection of high-level errors: more costly to fix; additional cost of writing drivers

Test plan, test data, test cases

test data: a set of inputs devised to exercise a test

test case consists of

1. the purpose of the test in terms of the system requirements it exercises
2. an input specification (ie test data)
3. a specification of the expected output

test plan: the major components are

- a description of the major phases of testing (eg unit testing, module testing, ...)
- objectives (acceptance criteria) for the testing process
- an overall testing schedule and resource allocation (when, who, time and machine resources)
- a description of the relationship between the test plan and other project plans (eg implementation schedule)
- a description of how traceability of test results to system requirements is to be demonstrated
- a description of how tests results are recorded (It must be possible to audit the testing process to guarantee that tests have been carried out on latest versions of the software.)
- a description of how the test cases were designed, and how the test data was generated
- a description of all the test cases, including all test data

The test schedule should allow for slippage