Software Test Plan

By

Bugbusters
Outline

- Test Phases
- Overhead Software, Environment & Resources
- Unit Testing
- Integration Testing
- Validation Testing
- System Testing
- Schedule
Test Phases

- Unit Testing
- Integration Testing
- Validation Testing
- System Testing
Overhead Software, Environment & Resources

- Test drivers and stubs in Java.
- Attempt to keep them simple to avoid high actual overhead.
- No off-the-shelf testing tools used since no additional advantage to be gained.
Unit Testing

- Heavy white-box testing techniques used to ensure maximum coverage and error detection.
- Involves conduction of low-order tests.
- Individual unit testing done by developers.
- Later, each developer closely works with one or more independent testers to ensure efficiency.
Integration Testing

- Integration testing also left to developers, later to be taken over by independent testers.
- Focus on construction of the software architecture.
- Black-box techniques used mostly.
- Testing with relatively high-order tests.
Integration Testing (Contd.)

- Critical modules identified.
- Incremental integration followed by regression testing.
Testing Strategy

• Moving spirally outwards from unit to system testing.

• Top-down approach
  – early detection of major control problems important.

• Developers provided with a set of general guidelines for unit testing involving:
  – interface testing
Testing Strategy (Contd.)

- data structure testing
- boundary value conditions
- error handling while debugging
Builds

- User Interface
- Directed Graph
- Test Case Generator
User Interface

TSG instantiates Main Application Frame which shows Create Structure Dialog. Create Structure Dialog shows Select Data Type Dialog, which shows Adequacy Criteria Dialog and Template Data Type Dialog. Template Data Type Dialog shows Methods Parameters Dialog. Output Wrapper shows Configuration Object.
User Interface Testing

- Usability Testing
- Test Control Flow
- Use wrapper to test outputs
Directed Graph Generator

User Interface Build → Configuration Object

Internal Representation creates

Output Stub

Directed Graph
Directed Graph Testing

- Use test internal representation files
- Test automatic and manual directed graph generation
- Check directed graph
Test Case Generator

Directed Graph Build creates Test Frame Manager

Test Frame creates Test Case Manager

Test Scripts creates Test Case Manager

Test Case Manager uses Value Generator
Test Case Generator

- Use same internal representation files used for directed graph.
- Set varying data types and boundary values.
- Use system to create test scripts, and check them for errors.
Validation Testing

• Ensure that final product meets specifications.
• Test final product against use cases set out in SRS.
Validation Testing Example

Generate test cases for the first time

1. Select Generate test cases for the first time.
2. Select the internal representation file of the software component from
3. Choose test adequacy criteria from a list.
4. Specify data type for template.
5. Specify boundary values for templates.
6. Specify the links between the nodes of the directed graph that cannot be generated automatically.
7. Add comments to the test script using the editor after test script generation completed.
System Testing

• Stress Testing (Sensitivity testing)
  – extreme boundary values
  – complicated data structure, with excessive control flow
• “finger pointing” handling.
• Porting to Unix
Schedule

- 23 NOV - Begin Unit Testing
- 29 NOV - Begin Testing UI Build
- 2 DEC - Begin Testing DG Build
- 4 DEC - Begin Testing TCG Build
- 7 DEC - Validation & System Testing
Questions?