

SOFTWARE TESTING

T8

200310548 이정우

200911364 곽수미

200911372 김민하

200911373 김바울

Contents

- **Introduction**
- **Types of tests**
- **Test Levels**
- **Testing Strategies**
- **Test Design**
- **Test Coverage**
- **Test Execution**
- **Test Documentation**
- **Test Management**

Introduction

□ Definition in IEEE Standard 610.12

- To conduct an activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.
- A set of one or more test** cases and/or procedures.

▪ “Test” means

- All of / A part of the activity
- All of / A part of the product the activity

Types of tests

- The SWEBOK provides a list of Types of tests.
 - The software engineer's intuition and experience
 - Specifications
 - Code
 - Dataflow
 - Fault
 - Usage
 - The nature of the application

Types of tests

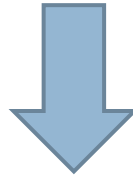
- The most commonly used types of tests.
 - Equivalence class partitioning
 - Boundary value
 - Decision table
 - Exploratory
 - Operational profile

Equivalence class partitioning

- To examine the defined acceptable range for each input to determine the following classes for each input.
 - Valid(s) : continuous range or list of values that should be legal and processed successfully by code.
 - Invalid(s) : continuous range or list of values that should be illegal and not accepted by the software, but not cause an acceptable result, either.

Equivalence class partitioning

- An example : The age of an individual
 - The valid range could be 0 to 120



- Equivalence classes :
 - Valid : {0-120}
 - Invalid : {<0} and {>120}

Equivalence class partitioning

□ The steps

- Define the valid and invalid class for each input.
- Test as many of the valid classes together as possible.
- Do one test for each invalid classes.

Boundary value testing

- Mandates the testing of four values for each input
 - The legally defined minimum
 - The legally defined maximum
 - The first possible value below the legally defined minimum
 - The first possible value above the legally defined maximum

Boundary value testing

- An example : The age of an individual
 - The legally defined minimum : 0
 - The legally defined maximum : 120
 - The first possible value below the legally defined minimum : -1
 - The first possible value above the legally defined maximum : 121
- It can be applied for the length of the field for nonnumeric inputs.

Decision table

- To cover all possible combinations of the input conditions in the tests.
 - Lists the all possible conditions and actions in the first column of a table
 - There is a column or “rule” for every possible combination of input conditions.
 - A particular input condition is marked with a Y for yes, an N for no, or an I for immaterial (both yes and no)

Decision table

- An example : For computing U.S. payroll tax withholding

Table 1

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Wages earned	N	Y	Y	Y
End of pay period	I	N	Y	Y
FICA limit exceeded	I	I	N	Y
Actions				
Withhold FICA tax	N	N	Y	N
Withhold Medicare tax	N	N	Y	Y
Withhold payroll tax	N	N	Y	Y

Exploratory testing

- Not require preplanning of exact data values
- Planning the focus the testing process
- Not mandatory to stick to the plan
- The tester reacts to the results being produced and modifies the plan accordingly.
 - For example, if many problems are being discovered with a particular aspect of the system, more tests are created for that area.

Operational profile

- To test more for the features that are used more.
 - As a result, achieve more robustness of the software
 - Recommends that the number of tests run for each system feature follow the model of how much it is used during operation
 - Actual usage may be measured
 - Useful for a regression test

Test levels

- Testing is done at more than one level.
 - Because, a software product is being developed or maintained
- Testing levels vary
 - To the scope of software being tested
 - The test techniques, objectives, environment
- Testing level schemes start with the smallest scope to be tested and increase in scope.

Test levels

- Some organizations also have multiple levels of test for the entire system product.
 - For example, security, performance, usability etc.
- The number and types of test levels vary from organization to organization and even with projects within one organization.
- Any one of these levels may have multiple level within it.

Test levels

- Main factors that effect the number and types of levels
 - System size
 - Complexity
 - Safety criticality
 - The experience of the testing staff and management
 - The degree of desire for certification
- Each organization makes its own decision about how many levels of test to have, and improves it if the goals of the organization are not being met.

Test levels

- Definition of test levels in IEEE/EIA 12207.0.
 - Different processes have different testing level needs.
- The acquisition process
 - The acquirer checks that all of the predefined acceptance conditions are met

Test levels

- The development process
 - Each software unit and database
 - Integrated units and components
 - Tests for each software requirement
 - Software qualification testing for all requirements
 - System integration
 - System qualification testing for system requirement

Test levels

- The operation process
 - operational testing
- The maintenance process
 - All development levels for improvements and adaptation
 - Test the modified parts of the system
 - Test the unmodified parts
 - Migration verification
 - Possibly also parallel testing

Test levels

- The supporting process
 - Verification
 - The software products of a specific activity successfully implements the requirements of the immediately prior activity
 - Validation
 - The final, as-built software or system product meets all of the requirement

Test levels

- Definition test levels in The SWEBOK
 - Unit testing verifies the functioning in isolation of software pieces which are separately testable.
 - Integration testing is the process of verifying the behavior of a whole system.
 - System testing is concerned with the behavior of a whole system.

Testing strategies

- Reflects the priorities of the organization
- Leads to the emphasis and coverage goals for the test cases themselves
- Focused on either “macro” or “micro” test issues

Test strategy issues - Macro

- Time-to-market
 - Speed of test development and execution

 - Amount of functionality to be delivered
 - And as a result, tested

 - Quality of the product
 - thoroughness of the testing
- >These three issues are trade-offs.

Test strategy issue - Macro

- Most organizations want all three.
 - > Therefore, Strategy creates priorities for the selection and management of all test activities.

Test strategy issue - Macro

- Another “macro”

- The desire for the cost of all development and maintenance of all test activities to meet expectations.
- Role in supporting the organization’s business goals

Test strategy issues - Micro

- Focused on test processes and products

- Some examples (support “macro” strategies)
 - For reduction of time to market
 - More automation to speed test execution
 - Fewer turnovers in testing staff to minimize time for learning curves
 - Better selection of the test cases that are executed, to find the more serious problems earlier

Test strategy issue - Micro

- For functionality changes
 - Tracing from requirements to test cases to enable finding the cases affected by the changes quickly
 - Smaller, more modular test scripts to maximize reuse during changes in functionality
- For quality of the product
 - Better tools to measure current test coverage
 - Better unit test tools for developers
 - More variety in the test types

Test strategy issue - Micro

- To control test effort costs
 - Use a project management tool to estimate test activities and to track the actual expenditures
 - Add root cause analysis(RCA) to test readiness review meeting

Test design

- Test design = Art + Science
- The goal of test design
 - To get the most return with the least effort!
- A good test design includes
 - Both structured and unstructured techniques

Unstructured techniques

- Random
 - Ad hoc
 - Performed without planning and documentation
 - A part of exploratory testing
 - Intended to be run only once, unless a defect is discovered
 - Exploratory
- Advantage
- Find lots of problems, often quite serious

Structured techniques

- Equivalence class partitioning
 - Boundary value
 - Decision table
- Advantage
- Provide linear coverage
 - All attributes are tested to the same degree in exactly the same manner.

Test coverage of code

- Measuring how much a software program has been exercised by tests(code).
- The goal for coverage of code during testing will vary with the level of test.

Test coverage of code

- Ways to define – code was “Covered”
 - To count it as covered if it was executed in any part
 - Count physical line of code of logical line of code
- The most robust definition is to use logical lines of code with 100% coverage.
 - To call out a need for execution of every condition
 - Calls for 100% of all possible paths for execution

Test coverage of code

- A universally recognized technique for achieving 100% coverage.
 - > Tom McCabe's Basis Path Testing technique
 - The goal is 100% condition coverage, but not every possible path.

Tom McCabe's Basis Path testing

□ The steps

- Draw a flowgraph
 - Each logical statement (node) : circle
 - The transfer of control as a result of decisions (edges) : arrows
- Compute the metric cyclomatic complexity
 - Count the number of regions
 - number of edges – number of nodes + 2
- Choose the paths

Table 2

Code statements (in simplistic pseudocode)	Corresponding flowgraph
001 If A 002 then B 003 End if 004 For 1 to n, do 005 If more data, 006 Then add to total C 007 Else send error message D 008 End if 009 End for	
The Cyclomatic Complexity is equal to the number of regions, which is 4.	
Four possible test cases would follow these paths: 001,003,004,005,006,008,009 001,002,003,004,005,008,009 001,003,004,005,007,008,009 001,003,004,005,006,008,009,004,005,007,008,009	
In this case, Cyclomatic Complexity could be reduced to Actual Complexity of 2, with the following paths: 001,003,004,005,006,008,009 001,002,003,004,005,007,008,009,004,005,006,008,009	

Tom McCabe's Basis Path testing

- Actual complexity
 - To test all of the nodes and edges with fewer than the cyclomatic complexity number of paths.
- Cyclomatic complexity
 - Included on certification exams in one form or another
 - One of the few aspects of software engineering that produces a numerical answer.
 - provides an easily deterministic exam question and answer

Test coverage of specifications

□ Traceability matrix

- If the contents of a specification have been inventoried in some manner.
- After completed, each requirement can be traced to the test cases where it is exercised by filling in a column in the table identifying these cases.
- The tracing has been completed, it is possible to compute the % requirements tested or a similar metric.

Table 3

ID #	Requirement summary	Test scenario/cases
001	Selection of items for purchase.	PUR002/35-40, 66-85 PUR003/1-57
002	Entry and validation of credit card information	CRC001/1-36 CRC002/5-89 CRC003/5-34, 75-115
003	Limit the maximum items purchased at one time as a fraud countermeasure	PUR001/56-77 PUR002/41-65

Test execution

- The needs for the test execution are specified in the Test Plan.
 - Specifies every component of the test environment, hardware, software, automated tools, data, personnel
- All of the actual test input and procedure are specified in test cases and test procedures.
 - Need to be documented with enough clarity
 - Another individual can replicate the results

Test execution

- The test results are logged during execution following the test plan, including an evaluation as to the success or failure of each test cases.
- Incident reports are recorded during test execution to allow both the developer and subsequent tester of the repaired code to reproduce the original problem.

Test execution

- Testing may not go according to plan.
 - Test execution is adapting to unplanned changes
- “Test Director” or “Test Coordinator”
 - To adapt to the changes from the plan
 - To redirect the total test execution effort
- When the test execution is completed
 - the results are documented in the test summary report.

Test documentation

- Test documentation is recorded in media.
 - Example : word processor, databases etc.
- The media selected for documentation will vary.
 - The level of detail of the test document
 - The experience of the testers preparing and using the documentation
 - The availability of automated test tools

Test documentation

- Standard for Software Test Documentation
 - Test plan : The overall resources, test environment scope of what is to be tested/not tested, methods
 - Most organizations do
 - Includes management planning information
 - Covered in an overall project plan
 - Skipping the test plan also works well when the test processes and environments are stable from one software release to the next.

Test documentation

- Test design : A more detailed level of methodology
 - Information for an identified subset of the overall scope
- Test cases : The actual data needed to run the test
- Test procedures : The steps for the pretest setup, test execution, posttest activities
- Test logs : The actual test results
- Incident reports : Descriptions of test result that do not match expectations
 - Incident reports are virtually always tracked with a database.

Test documentation

- Test summary report : The pass or fail decision for the test, the rationale for that decision, a summary of all test results, and the detailed test results
- Test item transmittal report : An inventory of all test documents and data being delivered as a result of a test.

Test documentation

- The documentation requirements are tailored by individual organizations to better meet their needs and abilities.
 - To combine one or more of the documents
- The trade-offs of time-to-market vs. quality vs. completeness of the product are discussed, and a decision is made as to what software can go into production when.

Test management



- Includes all of the normal project management activities for the test aspects of a project
 - Estimating schedules
 - Planning for staffing and training
 - Identifying and planning tasks
 - Monitoring the execution of the plans and replanning based on the results

Test management

- A standard management techniques that is particularly needed in testing is the concept of management reserve.
 - This is where a manager has resources that are available, but held back and not allocated until something goes wrong with the existing plans.
 - Metrics and measurement programs are a useful tool for test management.