

FUNDAMENTALS OF SOFTWARE TESTING

INTE-E3 – SYSTEMS TESTING AND AUTOMATION

Topic Outline

- Understand the fundamentals of software testing and the importance of testing in determining high quality software.
- Understand the importance of the mindset of a software tester and the influence of a test manager.
- Identify the available test tools for support testing and their benefits.

Introduction

Testing plays a key role in verifying the correctness of software and confirming that the requirements have been correctly implemented.

It is a **constructive** and **destructive** activity in that while, on the one hand, it aims to verify the correctness of the software, on the other hand, it aims to find as many defects as possible in the software.

The vast majority of defects (e.g. 80%) are detected by software inspections in a mature software organization, with the remainder detected by the various types of testing carried out during the project.

Introduction

Software testing involves defining the test conditions and designing the testcases and then executing the test cases.

Several types of testing are performed during the project, including ***unit, integration, system, regression, performance,*** and ***user acceptance testing.***

White box testing is based on knowledge of the internals of the software module.

Black box testing is to verify the functionality of a module (or feature or the complete system itself), and knowledge of the internals of the software module is not required.

Software Test Process

- Test planning and risk management
- Dedicated test environment and test tools
- Test case definition
- Test automation
- Test execution
- Formality in handover to test department
- Test result analysis
- Test reporting
- Measurements of test effectiveness
- Test management
- Lessons learned and test process improvement.

Software Test Process cont.

Test planning consists of a:

- Documented plan defining the scope of testing

- Various types of testing to be performed

- Definition of the test environment

- Required hardware or software for the test environment

- Estimation of effort and resources for the various activities

- Risk management

- Deliverables to be produced

- Key test milestones

- Test schedule

Types of Testing

Test Type	Description
Unit testing	This testing is performed by the software developers to verify the correctness of the software modules
Component testing	This testing is performed by the software developers to verify the correctness of the software components, i.e. to ensure that the component is correct and may be reused
System testing	This testing is generally carried out by an independent test group to verify the correctness of the complete system
Performance testing	This testing is generally carried out by an independent test group to ensure that the performance of the system is within the defined parameters. It may require tools to simulate clients and heavy loads, and precise measurements of performance are made
Load/stress testing	This testing is used to verify that the system performance is within the defined limits for heavy system loads over long or short periods of time
Browser compatibility	This testing is specific to Web-based applications and verifies that the website functions correctly with the supported browsers

Types of Testing cont.

Test Type	Description
Usability testing	This testing verifies that the software is easy to use and that the look and feel of the application is good
Security testing	This testing verifies that the confidentiality, integrity, and availability requirements are satisfied
Regression testing	This testing verifies that the core functionality is preserved following changes or corrections to the software. Test automation may be employed to increase its productivity and efficiency
Test simulation	This testing simulates part of the system where the real system currently does not exist, or where the real-life situation is hard to replicate
Acceptance testing	This testing carried out by the customer to verify that the software is fit for purpose and matches the customer's expectations

Test Levels

Test level	Description
Component testing	Each component is tested separately prior to integration with others. It may include functional, non-functional, and structural tests. The test cases are based on software design and code structure
Integration testing	The integrated components are tested together with functional, non-functional, and structural tests
System testing	The integrated system is tested by a dedicated test team using functional and non-functional tests (sometimes structural testing—e.g. page navigation)
Acceptance testing	This testing is the responsibility of the customer, and the goal is to verify that the software is fit for purpose and matches the customer's expectations. It involves user acceptance testing and operational acceptance testing. It may involve contractual and compliance acceptance testing and alpha/beta testing

Software Test Planning and Scheduling

Testing is a subproject of a project and needs to be managed as such, and so good planning and monitoring and control are required.

The **IEEE 829 standard** includes a template for test planning:

- defining the scope of the testing to be performed;

- defining the test environment

- estimating the effort required to define the test cases and to perform the testing

- identifying the resources needed (including people, hardware, software, and tools)

- assigning people to the tasks

- defining the schedule

- and identifying any risks to the testing and managing them

Simple Test Schedule

Activity	Resource name(s)	Start date	End/replan date	Comments
Review requirements	Test Team	15.02.2019	16.02.2019	Complete
Project test plan and review	J. DiNatale	15.02.2019	28.02.2019	Complete
System test plan/review	P. Cuitino	01.03.2019	22.03.2019	Complete
Performance test plan/review	L. Padilla	15.03.2019	31.03.2019	Complete
Regression plan/review	X. Yun	01.03.2019	15.03.2019	Complete
Set-up test environment	X. Yun	15.03.2019	31.03.2019	Complete
System testing	P. Cuitino	01.04.2019	31.05.2019	In progress
Performance testing	L. Padilla	15.04.2019	07.05.2019	In progress
Regression testing	L. Padilla	07.05.2019	31.05.2019	In progress
Test reporting	J. DiNatale	01.04.2019	31.05.2019	In progress

Test Case Design and Definition

The **unit tests are based on the software design**; the **system tests are based on the system requirements**; and the **UAT tests are based on the business** (or user) requirements.

The **unit tests are more focused on white box testing**, whereas the **system test and UAT tests are focused on black box testing**.

A test script generally includes:

- Test case ID
- Test type (e.g. unit, system, UAT)
- Objective/description
- Test script steps
- Expected results
- Actual results
- Tested by

Test Execution

The test group will usually be **independent** (i.e., it has an independent reporting channel from the development manager), and the test activities will usually include system testing, performance testing, usability testing, and so on.

There is usually a **formal handover** from development to the test group prior to the commencement of testing,

And the handover criteria need to be satisfied in order for the software to be accepted for testing by the test group.

Test Reporting and Project Sign-Off

The test manager will report progress regularly during the project.

The report provides the current status of testing for the project including:

- Quality status (including tests run, passed, and blocked).

- Risks and issues

- Status of test schedule

- Deliverables planned (next period)

Testing and Quality

Testing allows the quality of the software to be measured in terms of the defects found.

It provides confidence in the quality of the software, and a properly designed test case that passes reduces the level of risk with the system.

Further, the quality of the system generally increases (***reliability growth***) once defects identified during testing have been corrected and verified.

What is Software Defect?

A **defect** is a flaw in the software that causes the software to fail to perform its required function.

A defect that is encountered during program execution leads to a software failure.

A defect arises due to a developer making an error that produces a defect in the **code**.

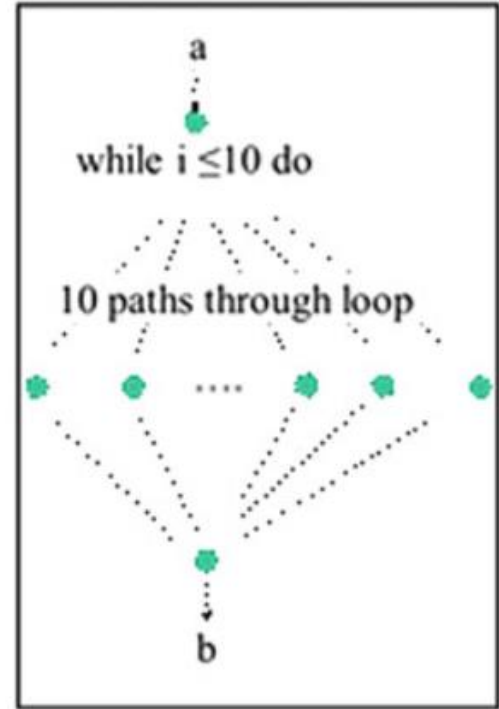
The code where the defect is present may be executed, which results in the software failing to do what it is required (i.e. failure).

The defect density of the software is the number of defects in the software divided by the number of lines of code.

Is Exhaustive Testing Possible?

Exhaustive **black box testing** would involve using every possible test input condition (valid and invalid), and the number of test cases rapidly becomes astronomical (**potentially infinite**).

Similarly, it is unrealistic in **white box testing** to perform exhaustive path testing and to execute all possible paths through the program.



How Much Testing Should Be Done?

The amount of software testing to be done ***depends on the level of business and technical risks***, as well as the project risks and the constraints on time and budget.

Testing should ***provide sufficient information*** to allow the stakeholders to make informed decisions as to whether further testing should be done or whether the product is ready to be released.

Testing and Quality Improvement

Test defects are valuable in the sense that they provide valuable information that allows the organization the opportunity to improve its software development process to ***prevent the defects from reoccurring in the future.***

A mature development organization will perform internal reviews of requirements, design, and code prior to testing.

Psychology of Software Tester

Myers observed that the psychology of the person carrying out the testing influences the testing (**Myers 1979**).

If the tester believes that the purpose of the testing is to demonstrate that the software works correctly (**constructive mindset**), then the tester is likely to focus on proving this point and using inputs for which correct results will be obtained.

Similarly, if the tester believes that the purpose of testing is to show that the software does not work (**destructive mindset**), then this approach helps in identifying most of the defects in the software.

E-commerce Testing

A **website** is a software application and so standard software engineering principles are employed to verify its quality.

E-commerce applications are characterized by:

- Distributed system with millions of servers and billions of participants

- High availability requirements (24 * 7 * 365)

- Look and feel of the website is highly important

- Browsers may be unknown

- Performance may be un-predictable

- Users may be unknown

- Security threats may be from anywhere

- Rapidly changing technologies.

E-commerce Testing cont.

Rapid application development model such as RAD/JAD or Agile is employed to design a little, implement a little, and test a little, and the standard waterfall lifecycle model is rarely employed for the front end of a Web application.

The various types of **Web testing** include:

- Static testing

- Unit testing

- Functional testing

- Browser compatibility testing

- Usability testing

- Security testing

- Load/performance/stress testing

- Availability testing

- Post-deployment testing

E-commerce Testing cont.

Static testing of websites is to check the content of the Web pages for accuracy, consistency, correctness, and usability and also to identify any syntax errors or anomalies in the HTML.

Unit testing is to verify that the content of the Web pages corresponds to the design, that the content is correct, that all the links are valid, and that the Web navigation operates correctly.

Functional testing is to verify that the functional requirements are satisfied.

Browser compatibility testing is to verify that the Web browsers that are to be supported are actually supported.

Usability testing is to verify that the look and feel of the application is good and that Web performance (loading Web pages, graphics, etc.) is good.

Security testing is to ensure that the website is secure.

The **purpose of load, performance, and stress testing** is to ensure that the performance of the system is within the defined parameters.

Post-deployment testing is to ensure that website performance remains good following deployment at the customer site, and this may be done as part of a service-level agreement (SLA).

Software Maintenance and Evolution

Software maintenance is the process of changing a system after it has been delivered to the customer, and it involves correcting any defects that are present in the software and enhancing the system to meet the evolving needs of the customer.

The defects may be due to coding, design, or requirements' errors, with coding defects less expensive to fix than requirements' defects.

Software systems often have a long lifetime (**e.g., some systems have a lifetime of 20-30 years**), and so the software needs to be continuously enhanced over its lifetime to meet the evolving needs of the customer.

Maintenance testing plays a key role in verifying that the new release is fit for purpose.

Software Test Tools

Test tools are employed to support the test process and to **enhance quality and increase productivity**.

Tool selection needs to be planned, and the selection of a particular tool involves defining the requirements of the proposed tool and identifying candidate tools to evaluate against the requirements.

There are various tools to support testing such as **test planning and management tools; defect tracking tools; regression test automation tools; performance tools**; and so on.

There are tools available from various vendors such as Compuware, Software Research, Inc., HP, LDRA, McCabe and Associates, and IBM Rational.

Software Test Tools cont.

Test Management Tools

There are various test management tools available and their main features are:

- Management of entire testing process

- Test planning

- Test status and reporting

- Graphs for presentation

- Defect tracking system

- Support for many testers

- Audit trail proof that testing has been done

- Test automation

- Support for various types of testing.

Software Test Tools cont.

