University of Tripoli – Faculty of Information Technology

Software Engineering Department

Software Quality Assurance
ITSE421

Spring 2024

By Marwa Solla

Software Quality Assurance and Testing

Lecture 4 : Software Testing Fundamentals

Tripoli University · Faculty of Information Technology

Marwa Solla

m.solla@uot.edu.ly

What We Learn In This Lecture

o   What is Software Testing ?

o   Categories of testing

o   *Verification and Validation*

What is software testing

Software Testing

- It is process to test an application to find out error in it. Checking the software is ok.

- The goal of software tester to find bug.

- verifying and validating that a software or application is bug free

The goal of a software tester is to find bugs, find them as early as possible, and make sure they get fixed.
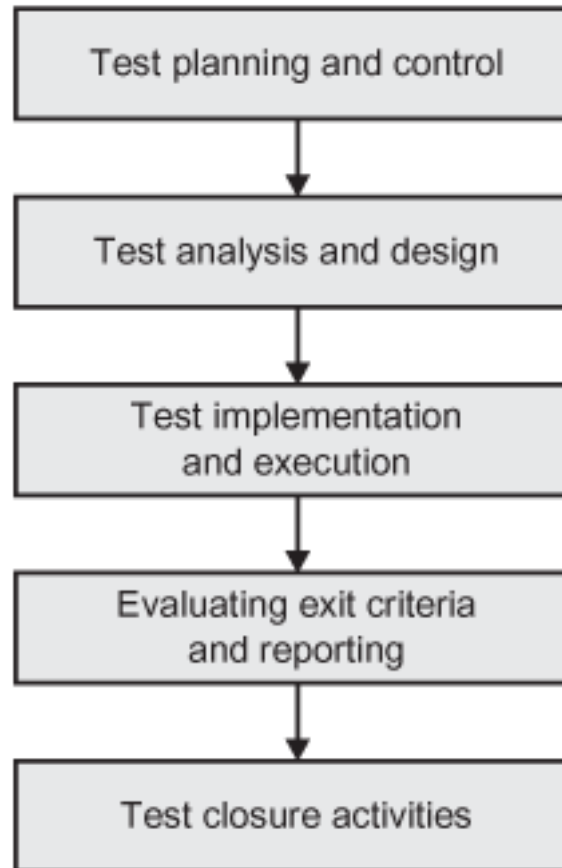
SOFTWARE TESTING

What is software testing

software testing is a *process* which includes many different activities; test execution (including checking of results) is only one of these activities. The test process also includes activities such as test planning, analyzing, designing, and implementing tests, reporting test progress and results, and evaluating the quality of a test object [1].

The testing process

Planning, design and performance of testing are carried out throughout the software development process

Software Testing

There are two main categories of testing:
1) **Static Testing**
2) **Dynamic Testing**

- *Static testing* : testing does not involve the execution of the component or system being tested. So, testing also includes reviewing work products such as requirements, user stories, and source code.

- *Dynamic testing*: testing does involve the execution of the component or system being tested;

Software Testing

Static Testing	Dynamic Testing:
Involves reviewing documents and code without executing the code (e.g., code reviews, walkthroughs	Involves executing the code and analyzing the output (e.g., functional and non–functional testing).
It is completed without executing the program.	It is completed with the execution of program.
This testing is executed in verification stage	This testing is executed in validation stage.
This testing prevents the defects.	This testing finds and fixes the defects.
The cost is less for finding the defects and fixes.	The cost is high for finding and fixing the defects.
It consists of Walkthrough, Inspection, reviews.	It consists of specification based, Experience based, unit testing, integration testing etc.

Verification and Validation

- Verification is the process confirming that something—software—*meets its specification*

- Validation is the process confirming that it meets the *user's requirements*

- Verification = meets specification as document

- Validation = meets User requirement (because software is ready hence main tested for User)

- Verification : Its static process of analyzing the document , not actual product

- Validation : it involves Dynamic Testing (unit, integration ,system testing)

Verification and Validation

Verification

- Verify the intermediary products like requirement documents, design documents, ER diagrams, test plan and traceability matrix
- Developer point of view
- Verified without executing the software code
- Techniques used: Informal Review, Inspection, Walkthrough, Technical and Peer review

Validation

- Validate the final end product like developed software or service or system
- Customer point of view
- Validated by executing the software code
- Techniques used: Functional testing, System testing, Smoke testing, Regression testing and Many more

Verification and Validation

Aspect	Verification	Validation
Purpose	Ensures the product is built correctly	Ensures the correct product is built
Focus	Processes and intermediate products	Final product and user needs
Activities	Reviews, inspections, walkthroughs	Testing, UAT, system testing
Techniques	Formal methods, static analysis	Black box testing, usability testing
Questions Answered	Are we building the product right?	Are we building the right product?

Typical Objectives of Testing

For any given project, the objectives of testing may include:

- To evaluate work products such as requirements, user stories, design, and code

- To verify whether all specified requirements have been fulfilled

- To validate whether the test object is complete and works as the users and other stakeholders expect

- To build confidence in the level of quality of the test object

- To prevent defects

- To find failures and defects

Typical Objectives of Testing

- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object
- To reduce the level of risk of inadequate software quality (e.g., previously undetected failures occurring in operation)
- To comply with contractual, legal, or regulatory requirements or standards, and/or to verify the test object's compliance with such requirements or standards
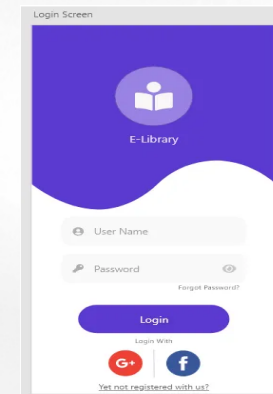
Typical Objectives of Testing

The objectives of testing can vary, depending upon the context of the component or system being tested, the test level, and the software development lifecycle model. These differences may include, for example:

• *During component testing,* one objective may be to find as many failures as possible so that the underlying defects are identified and fixed early. Another objective may be to increase code coverage of the component tests. (check: if else statements)


```
1   public void button1_Click(object sender, EventArgs e)
2   {
3       SqlConnection con = new SqlConnection(@"Data Source=USER;Initial Catalog=admin;Int
4       SqlDataAdapter sda = new SqlDataAdapter("SELECT COUNT(*) FROM login WHERE usernam
5       /* In the above line, the program is selecting the whole data from the table and
6       DataTable dt = new DataTable(); // Creating a virtual table
7       sda.Fill(dt);
8       if (dt.Rows[0][0].ToString() == "1")
9       {
10          /* I have made a new page called the home page. If the user is successfully a
11          this.Hide();
12          new home().Show();
13      }
14      else
15      {
16          MessageBox.Show("Invalid username or password");
17      }
18  }
```

• *During acceptance testing,* one objective may be to confirm that the system works as expected and satisfies requirements. Another objective of this testing may be to give information to stakeholders about the risk of releasing the system at a given time. (check validations)

14

Testing Types

Manual testing :– Manual testing includes testing a software manually, i.e., without using any automated tool or any script.

Automation Testing: Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re–run the test scenarios that were performed manually, quickly, and repeatedly

Levels of Testing

•**Unit Testing**: Focuses on individual components or units of a software.

•**Integration Testing**: Ensures that integrated components work together correctly.

•**System Testing**: Verifies the complete and integrated software system.

•**Acceptance Testing**: Confirms that the system meets the requirements and is ready for deployment.

Testing Techniques

• **Black Box Testing**: Testing without knowledge of the internal workings of the application.

• **White Box Testing**: Testing with knowledge of the internal logic of the system.

• **Gray Box Testing**: A combination of both black box and white box testing.
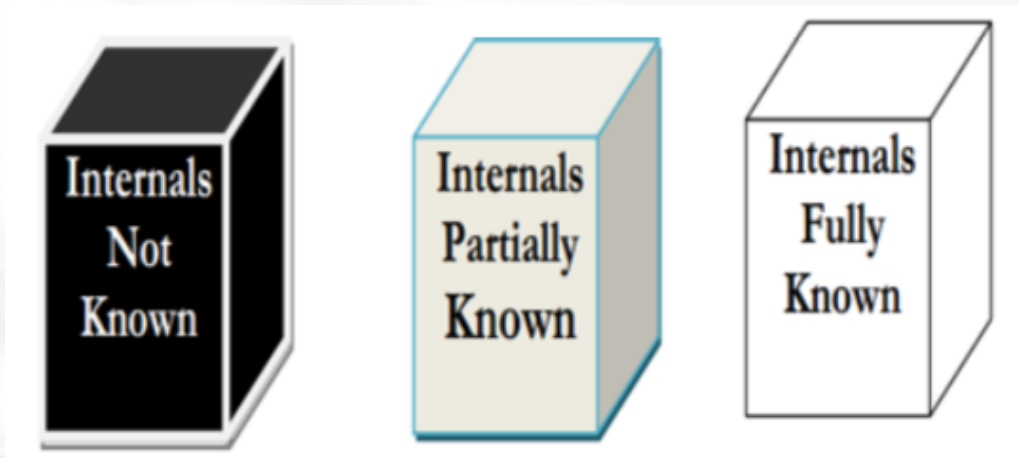
Seven Testing Principles

A number of testing principles have been suggested over the past 50 years and offer general guidelines common for all testing.

Principle 1. Early testing saves time and money

To find defects early, both static and dynamic test activities should be started as early as possible in the software development lifecycle. Early testing is sometimes referred to as *shift left*. Testing early in the software development lifecycle helps reduce costly changes.

Principle 2. Testing shows the presence of defects, not their absence

Testing can show that defects are present, but cannot prove that there are no defects. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, testing is not a proof of correctness.

Seven Testing Principles

Principle 3. Exhaustive testing is impossible

Testing everything (all combinations of inputs and preconditions) is not feasible except for trivial cases. Rather than attempting to test exhaustively, risk analysis, test techniques, and priorities should be used to focus test efforts.

Principle 4. Defects cluster together

A small number of modules usually contains most of the defects discovered during pre-release testing, or is responsible for most of the operational failures. Predicted defect clusters, and the actual observed defect clusters in test or operation, are an important input into a risk analysis used to focus the test effort (as mentioned in principle 3).

Seven Testing Principles

Principle 5. Beware of the pesticide paradox

If the same tests are repeated over and over again, eventually these tests no longer find any new defects. To detect new defects, existing tests and test data may need changing, and new tests may need to be written. (Tests are no longer effective at finding defects, just as pesticides are no longer effective at killing insects after a while.) In some cases, such as automated regression testing, the pesticide paradox has a beneficial outcome, which is the relatively low number of regression defects.
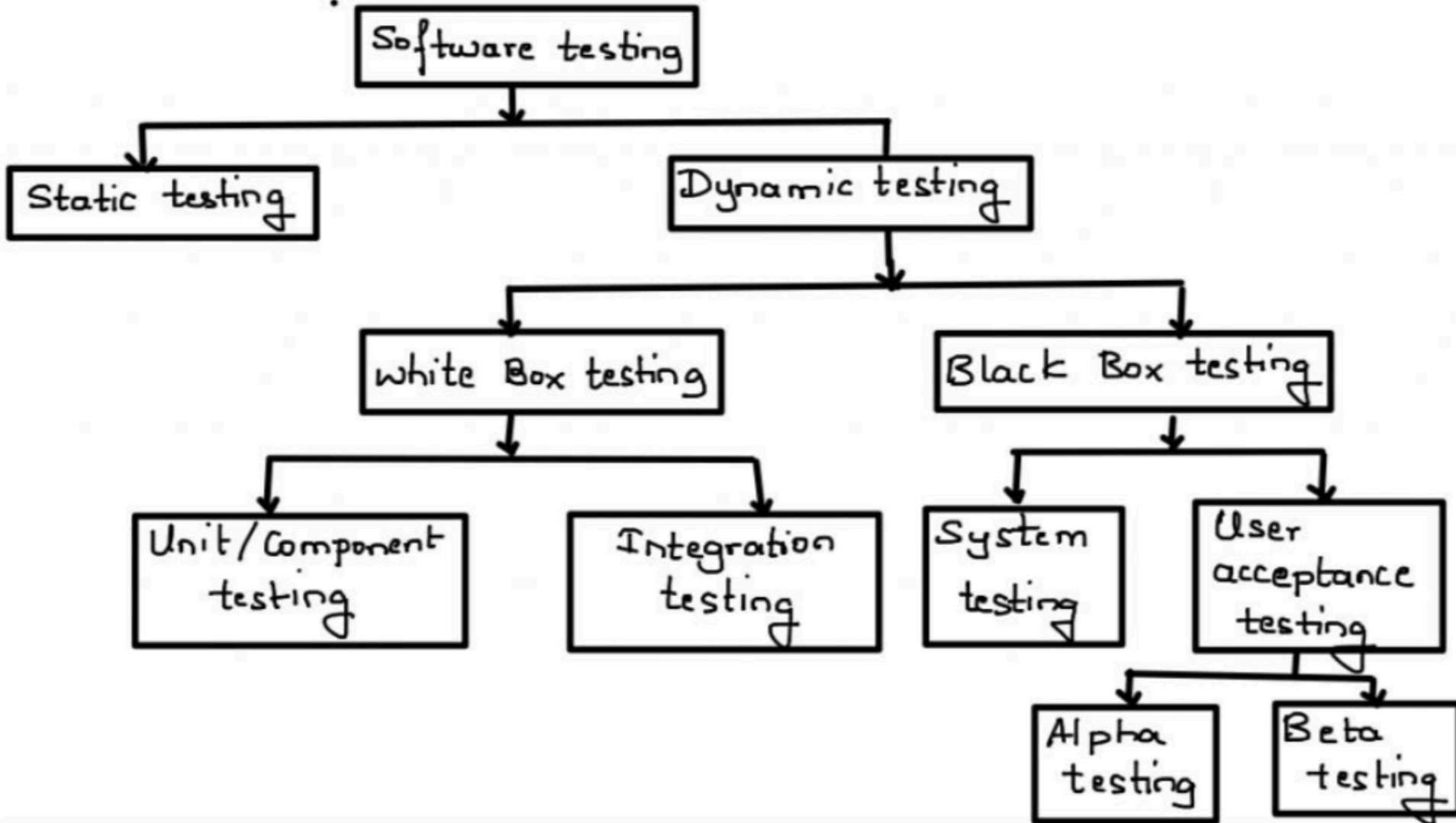
Seven Testing Principles

Principle 6.Testing is context dependent

Testing is done differently in different contexts. For example, safety-critical industrial control software is tested differently from an e-commerce mobile app. As another example, testing in an Agile project is done differently than testing in a sequential lifecycle project.

Principle 7- Absence of Error Fallacy

Finding and fixing defects does not help if the system built is unusable and does not fulfill the user's needs and expectations.

Software Testing

The end