

Software Testing I

Mojtaba Shahin

Week #9: Lecture – Part #1

Topics

- **Part 1**
 - Introduction to Software Testing
 - Terminologies in Software Testing

Notes and Acknowledgements

- Slides/images come from the following main sources:
 - **Chapter 8:** Ian Sommerville, Software Engineering, 10th Edition, 2015.
 - <https://iansommerville.com/software-engineering-book/slides/>
 - (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
 - <https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf>
 - Test Case vs Test Scenario: What's the Difference?
 - <https://www.guru99.com/test-case-vs-test-scenario.html>

Program testing goals

1. To demonstrate to the developer and the customer that **the software meets its requirements**.
 - For **custom software**, this means that there should be at least one test for **every requirement in the requirements document**.
 - For **generic software products**, it means that there should be tests for **all of the system features, plus combinations of these features**, that will be incorporated in the product release.
2. To discover situations in which the **behaviour of the software is incorrect, undesirable or does not conform to its specification**.

Validation and defect testing

- The first goal leads to **validation testing**
 - To demonstrate to the developer and the system customer that the software meets its requirements
 - A successful test shows that the system operates as intended.
- The second goal leads to **defect testing**
 - To discover faults or defects in the software where its behaviour is incorrect or not in conformance with its specification
 - A successful test is a test that makes the system perform incorrectly and so exposes a defect in the system.

Program testing

- When you test software, you execute a program using **artificial data**.
- You check the results of the test run for errors, anomalies or information about the program's non-functional attributes.
- Testing is part of a more general verification and validation process, which also includes static validation techniques.

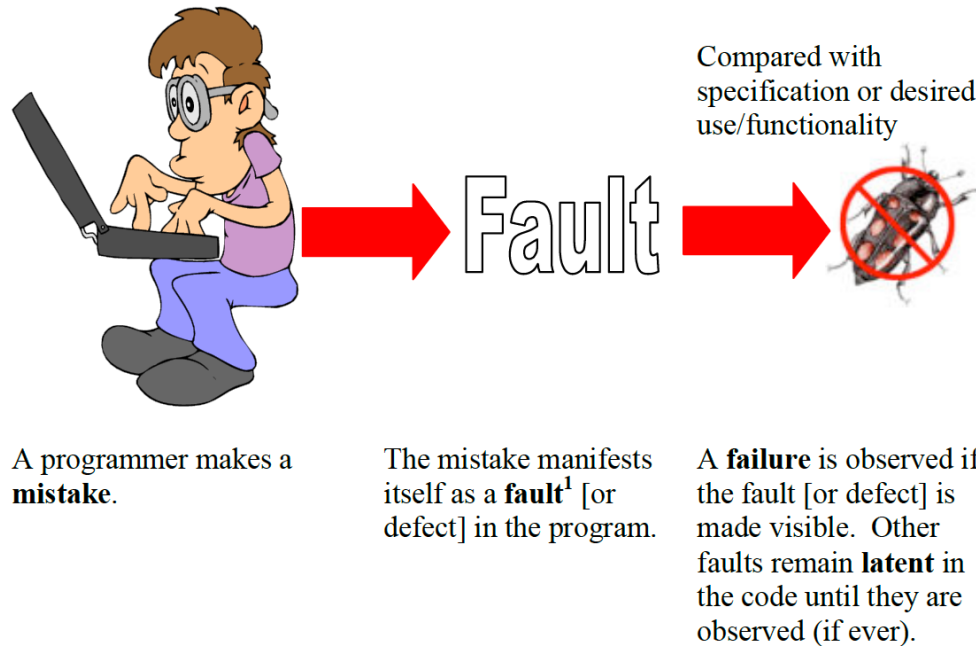
Testing Terminologies*

- **Mistake** – a human action that produces an incorrect result
- **Fault [or Defect]** – an incorrect step, process, or data definition in a program.
- **Failure** – the inability of a system or component to perform its required function within the specified performance requirement
- **Error** – the **difference** between a computed, observed, or measured value/condition and the true, specified, or correct value/condition.

Adopted from: (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
<https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf>

The progression of a software failure*

A **mistake** committed by a person becomes a **fault** (or **defect**) in a software artifact, such as the requirement specification, design, or code. This fault, unless caught, propagates as a defect in the executable code. When a defective piece of code is executed, the fault may become a visible anomaly and a **failure** is observed. Otherwise, the fault remains latent. **Testing can reveal failures, but it is the faults that must be found and removed.**



(Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
<https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf>

Verification vs Validation

- **Verification:**

"Are we building the product right".

—The software should conform to its specification.

- **Validation:**

"Are we building the right product".

—The software should do what the user really requires.

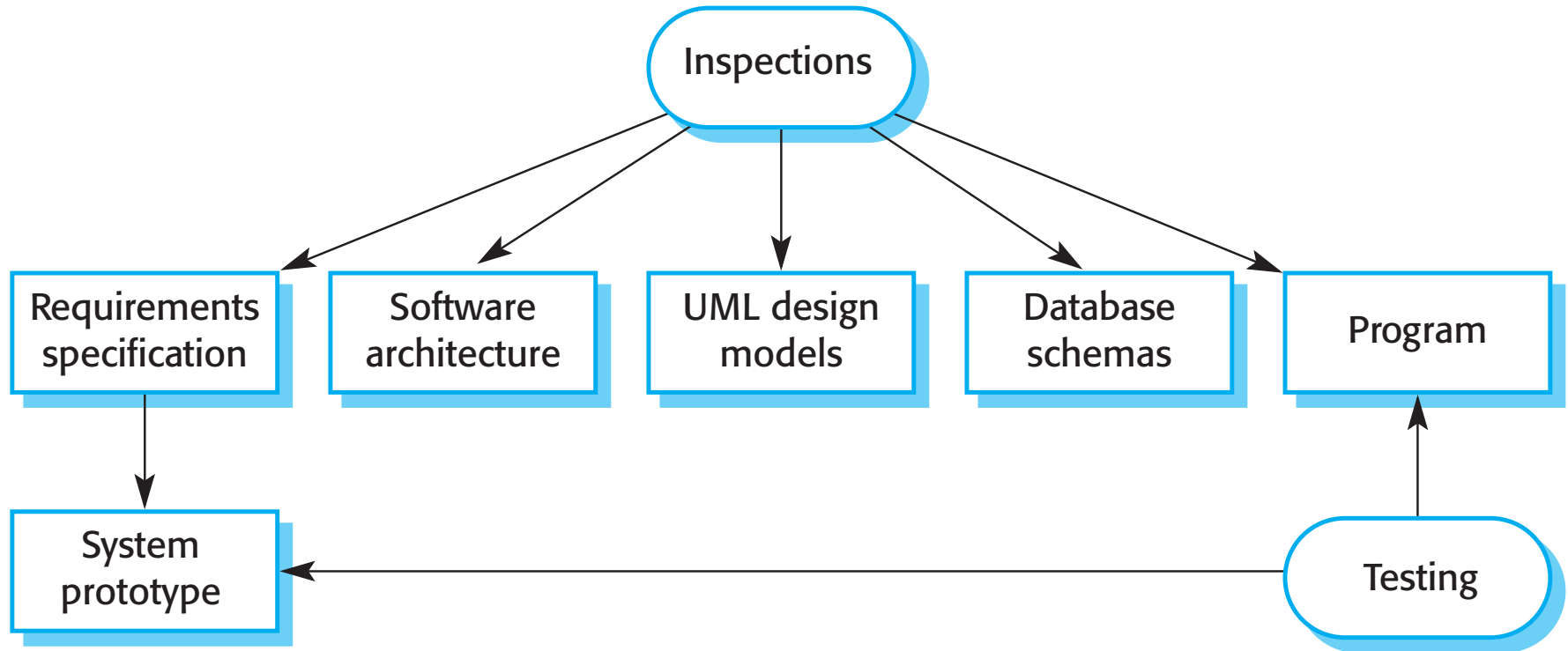
V & V confidence

- Aim of V & V is to establish confidence that the system is ‘fit for purpose’.
- The level of the required confidence depends on three factors
 - **Software purpose**
 - The level of confidence depends on how critical the software is to an organisation.
 - **User expectations**
 - Users may have low expectations of certain kinds of software.
 - **Marketing environment**
 - Getting a product to market early may be more important than finding defects in the program.

Software inspections

- It is concerned with the analysis of the static system representation to discover problems (**static verification**)
- Inspections do **not require execution of a system** so may be used before implementation.
- They may be applied to any representation of the system (requirements, design, configuration data, test data, etc.).
- They have been shown to be an effective technique for discovering program errors.

Inspections and testing



Advantages of inspections

1. During testing, errors can mask (hide) other errors. Because inspection is a static process, you don't have to be concerned with interactions between errors.
2. Incomplete versions of a system can be inspected without additional costs. If a program is incomplete, then you need to develop specialized test harnesses to test the parts that are available.
3. As well as searching for program defects, an inspection can also consider broader quality attributes of a program, such as compliance with standards, portability and maintainability.

Inspections and Testing

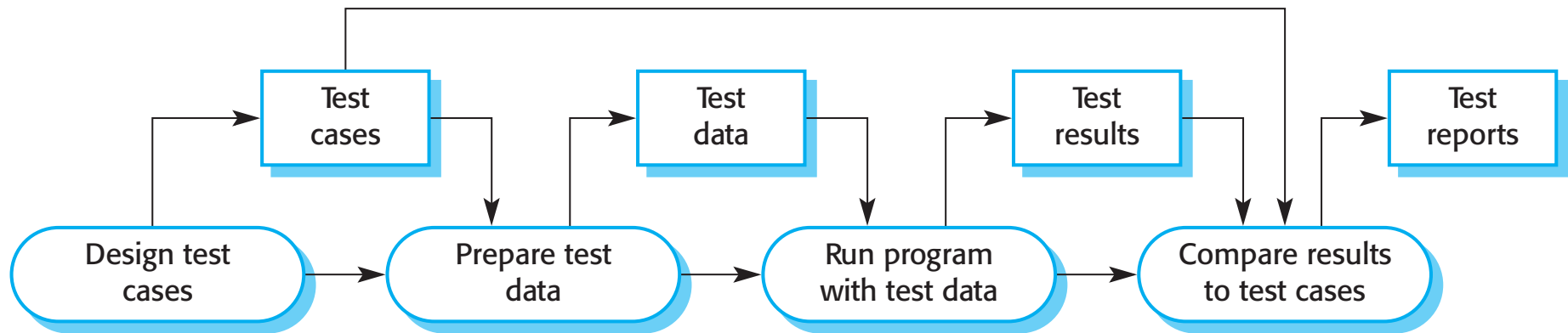
- Inspections and testing are **complementary** and not opposing verification techniques.
- Both should be used during the V & V process.
- Inspections cannot **check non-functional characteristics** such as performance, usability, etc.

Test Cases, Test Data, Test Results, Expected Test Results



- **Test cases** are specifications of the inputs to the test and the expected output from the system (the test results), plus a statement of what is being tested.
- **Test data** are the inputs that have been devised to test a system.
 - Test data can sometimes be generated automatically, but automatic test case generation is impossible.
- People who understand what the system is supposed to do must be involved to specify the **expected test results**.
- The **test results** are automatically compared with the predicted results, so there is no need for a person to look for errors and anomalies in the test run.

A model of the software testing process



Example of Test Cases

Test Scenario	Test Cases	Test Data	Expected Result	Test Result	Pass or Fail

Example of Test Cases

Test Scenario	Test Cases	Test Data	Expected Result	Test Result	Pass or Fail
Check the functionality of the Password_Checking function	Test Case 1. Verify the function with an INCORRECT password	Password: ABCD1234	Password is Invalid	Password is Invalid	Pass

Example of Test Cases

Test Scenario	Test Cases	Test Data	Expected Result	Test Result	Pass or Fail
Check the functionality of the Password_Checking function	Test Case 1. Verify the function with an INCORRECT password	Password: ABCD1234	Password is Invalid	Password is Invalid	Pass
	Test Case 2. Verify the function with a CORRECT password	Password: ABCD1234@	Password is Valid	Password is Invalid	Fail

Manual Testing vs Automated Testing

- In practice, the testing process usually involves a mixture of manual and automated testing.
- **Manual testing:** a tester runs the program with some test data and compares the results to their expectations.
 - They note and report discrepancies to the program developers.
- **Automated testing:** the tests are encoded in a program that is run each time the system under development is to be tested.
 - This is faster than manual testing.

Black-Box Testing vs White-Box Testing*

- **Black-box testing** (also called functional testing) is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.
- **White-box testing** (also called structural testing) is testing that takes into account the internal mechanism of a system or component.

Adopted from: (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition

<https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf>

Stages of testing

- **Development testing**, where the system is tested during development to discover bugs and defects.
- **Release testing**, where a separate testing team test a complete version of the system before it is released to users.
- **User testing**, where users or potential users of a system test the system in their own environment.

References

- Ian Sommerville, Software Engineering, 10th Edition, 2015.
 - <https://iansommerville.com/software-engineering-book/slides/>
- (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
 - <https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf>

Thanks!

Mojtaba Shahin

mojtaba.shahin@rmit.edu.au