Testing Techniques and Methods SFWR ENG 3S03: Software Testing

Alicia Marinache

Department of Computing and Software, McMaster University Canada L8S 4L7, Hamilton, Ontario SE 3SO3::Testing

A. Marinache

Preliminaries

State-Based

Code Visibility-Base

System

Requirements-Based

System Execution

evel of Syste Detail

Examples

Acknowledgments: Slides adapted from Dr. R. Khedri and Dr. R. Paige. Material based on [Per00]





(Slide 2 of 41)

SE 3SO3::Testing

A. Marinache

Objectives

- Discuss main testing techniques and methods
- Understand their goals, and how and when to use them

Preliminaries

Preliminaries

ystem tate-Based

Code Visibility-Based

System
Requirements-

vstem Execution

Level of System



(Slide 3 of 41)

 Testing Methods: high level strategy for conducting testing

• Testing Techniques: procedures to design test cases

- Classified by:
 - System State
 - Code Visibility
 - System Requirements
 - Level of System Detail
 - System Execution
 - ...and more

SE 3SO3::Testing

A. Marinache

Preliminaries

tate-Based

Code Visibility-Based

System Requirements-

ystem Execution

Level of System Detail



(Slide 4 of 41)

Definition (Static)

Testing the code without executing it

Goals

- Detect Defects Early
- Ensure Compliance with Standards

SE 3SO3::Testing

A. Marinache

Preliminaries

System
State-Based

Code

/isibility-Based System

Requirements-Based

System Execution

Level of System Detail



(Slide 5 of 41)

Definition (Dynamic)

Testing the code by executing it

Goals

- Verify Functional Behavior
- Identify Runtime Issues
- Validate Performance and Usability

SE 3SO3::Testing

A. Marinache

Preliminaries

System State-Based

ode

Visibility-Based
System

Requirements-Based

System Execution

Level of System Detail

⇒System State-Based

(Slide 6 of 41)

SE 3SO3::Testing

A. Marinache

Preliminaries

System State-Based

> Code /isibility_Base

ystem lequirements

Level of System

xamples

Example: Static/Dynamic testing: come up with an example for testing

- Web Server
- Smartphone App
- A Satellite System

System State-Based System State-Based

(Slide 7 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

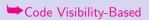
System State-Based

Code Visibility-Based

System Requirements-

System Execution

Level of System



(Slide 8 of 41)

Definition (White Box)

Testing that systematically exploits knowledge of the system to design tests for it

Sometimes, you don't have access to the source code:

- outsourcing or crowdsourcing the testing
- using a COTS product
- integrating an API

SE 3SO3::Testing

A. Marinache

Preliminaries

System
State-Rased

Code Visibility-Based

System Requirements-

ystem Execution

evel of System Detail

Code Visibility-Based →

(Slide 9 of 41)

Definition (Black Box)

Testing the system without knowledge of its internal design or implementation

Definition (Grey Box)

Testing the system with some knowledge of its internal structure and design

Examples

Black Box: COTS

• Grey Box: API testing, Web app

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-Based

System Execution

Level of Syster Detail



(Slide 10 of 41)

Goals

- Black Box:
 - Validate Requirements
 - Ensure User Satisfaction (Acceptance)
 - How: Test externally without code knowledge
 - When: functional and acceptance phases
- White Box:
 - Improve Code Quality
 - Ensure Code Coverage
 - How: Analyze internal logic
 - When: During development or for security and edge cases
- Grey Box:
 - Identify Integration Issues
 - Troubleshoot Complex Bugs
 - How: Combine both approaches
 - When: Integration and system-level

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code

Visibility-Based

System Requirements-Based

System Execution

evel of Syster etail

→Code Visibility-Based

(Slide 11 of 41)

Examples Black/Grey/White Box: come up with an example for testing

- Web Server
- Smartphone App
- A Satellite System

Hint: are they Static or Dynamic?

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

Visibility-Base System

Requirements Based

ystem Execution

Level of System Detail

→Code Visibility-Based

(Slide 12 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System
Requirements-

System Execution

Level of System

System Requirements-Based

(Slide 13 of 41)

Definition (Functional)

Testing that the system performs its function(s) correctly and the correctness can be sustained over a continuous period of time

- Goals: Ensure the software meets functional requirements; Identify functional defects
- How: Define test cases based on the functional specifications; Execute tests with input data to verify expected outputs
- When: During development and system testing: To validate individual features or modules; Before release: To ensure the system meets all functional requirements

SE 3SO3::Testing

A. Marinache

Droliminarios

Systom

Code Visibility-Based

System
Requirements-

System Execution

vel of Systen



(Slide 14 of 41)

Definition (Non-Functional)

Testing how the system performs (not what it does)

- Goals: Validate system quality attributes; Ensure a seamless user experience
- How: Simulate scenarios to measure non-functional aspects, like response time or resource usage; Use tools to automate tests and gather data
- When: Post-functional testing: To validate the system's readiness for deployment; During maintenance: To check for regressions in quality attributes after updates

SE 3SO3::Testing

A. Marinache

Droliminarios

Code

Visibility-Based

System

Requirements-Based

System Execution

evel of Syste Jetail

⇒System Requirements-Based

(Slide 15 of 41)

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-Based

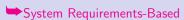
vstem Execution

Level of System Detail

xamples

Examples FT/NFT: come up with an example for testing

- Web Server
- Smartphone App
- A Satellite System



(Slide 16 of 41)

Functional Testing Techniques

• (Functional) Requirements testing

Acceptance testing

Regression testing

Error-handling testing

Manual-support testing

Integration testing

Parallel testing

SE 3SO3::Testing

A. Marinache

reminiaries

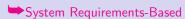
tate-Based

Code Visibility-Based

System Requirements-Based

ystem Execution

evel of System



(Slide 17 of 41)

Non-Functional Testing Techniques

- Stress testing
- Execution testing
- Recovery testing
- Operation testing
- Compliance testing
- Security testing

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System Requirements-Based

System Evesution

ystem Execution

. .

System Requirements-Based

(Slide 18 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System Requirements-

System Execution

Level of System

Examples

Based



(Slide 19 of 41)

Definition (Manual Testing)

Human testers executing test cases without the help of automation tools

- Goal: Identify any issues that the automated testing cannot; Validate usability and design
- Who: Testers (manual testing team); End-users
- How: Exploratory testing; Predefined test cases

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

iystem

Based

System Execution

Level of Systen Detail



(Slide 20 of 41)

Definition (Automated Testing)

Using tools and scripts to execute test cases automatically

- Goal: Ensure consistency in repetitive tasks; Increase speed of regression testing
- Who: Developers or automation engineers
- How: Test scripts using tools; Integration with testing frameworks

SE 3SO3::Testing

A. Marinache

_

System

Code Visibility-Based

System Requirements-

System Execution

Level of Systen

System Execution

(Slide 21 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

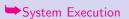
System

Code Visibility-Based

Requirements-Based

System Execution

Level of System Detail



(Slide 22 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System
Requirements-

Based

System Execution

Level of System Detail

```
Level of System Detail
```

(Slide 23 of 41)

Where do you start testing? One option

```
Method: Test:
int increment(int i){
    return i+1;
}
Test
@Test
public void test_1(){
    int j = 0;
    assertEquals(1, increment(j));
}
```

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System
Requirements-

ystem Execution

Level of System Detail

Level of System Detail

(Slide 24 of 41)

Definition (Unit Testing)

Testing individual components in isolation (to ensure they function as required)

Goal: Verify individual components

• Who: Usually done by the developer

When: Done during the development process

 How: Sometimes by creating the tests BEFORE the code (TDD) SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System
Requirements-

ystem Execution

Level of System

Level of System Detail

(Slide 25 of 41)

Challenges:

What is a "unit"?

• Can it catch all defects?

Solution...

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

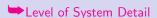
Code Visibility-Based

ystem

Requirements-Based

ystem Execution

Level of System Detail



(Slide 26 of 41)

Definition (Integration Testing)

The process that evaluates the interaction and data flow between integrated modules or components to ensure they work together as expected

- Goal: Detect integration defects
- Who: Could be engineers who have deep knowledge of the components
- How: Execute integration nodes
- When: Generally after modules have passed Unit Testing

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

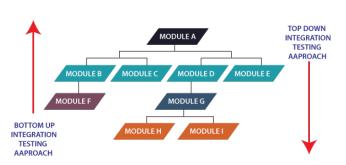
System
Requirements-

ystem Execution

Level of System Detail

Level of System Detail

(Slide 27 of 41)



Are we done yet? No, because...

(C)https://www.javatpoint.com/

SE 3SO3::Testing

A. Marinache

Proliminarios

....

Code

Visibility-Based

Requirements-Based

stem Execut

Level of System Detail

Level of System Detail

(Slide 28 of 41)

Definition (System Testing)

The comprehensive process that tests the system as a whole, to ensure it mets its specification as intended in real-world scenario

- Goal: Verify compliance with requirements
- Who: Usually expert team (not the developers)
- How: Run tests that cover the entire system and use realistic scenarios

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-

System Execution

Level of System Detail

Level of System Detail

(Slide 29 of 41)

Surely we are done now! Well...

Definition (Acceptance)

The final phase of testing, perform to validate the system against business requirements

- Goal: Business Approval
- Who: End-users, client representatives, or stakeholders
- How: Plan ... Execute ... Review Results

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-

System Execution

Level of System

Level of System Detail

(Slide 30 of 41)

You said FINAL, we ARE done, right?

Regression: "when you fix one bug, you introduce several newer bugs."









SE 3SO3::Testing

A. Marinache

Preliminaries

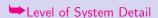
ystem

Code Visibility-Based

System Requirements

System Execution

Level of System



(Slide 31 of 41)

Definition (Regression)

Testing previously tested (segments of a) system to ensure they still function properly after a change has been made

 Goal: System documentation and test data remain current

Who: It depends on the change

• How: By re-running the original tests

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-

System Execution

Level of System Detail

JUnit Example of Regression Testing

Level of System Detail

(Slide 32 of 41)

SE 3SO3::Testing

A. Marinache

Preliminaries

stem ate-Based

Code

Visibility-Based

System Requirements-Based

ystem Execution

Level of System Detail

Examples

◆□ → ◆□ → ◆ □ → ◆

Level of System Detail

(Slide 33 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System
Requirements-

System Execution

Level of System Detail



(Slide 34 of 41)

For the following examples, think of as many **types** of tests you can perform

Hint: Always ask questions AND record your assumptions

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

> ystem lequirements-

ystem Execution

evel of System



(Slide 35 of 41)

Application: Calculator

Requirements:

- Perform basic arithmetic operations (+, -, *, /)
- Division should throw an error when dividing by zero

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System Requirements-

System Execution

Level of Systen



(Slide 36 of 41)

Application: Bank ATM Interface

Requirements:

- Withdraw funds if the balance is sufficient.
- Deny withdrawal for insufficient funds
- Allow balance inquiry and display correct amounts
- V2: Add Online banking

SE 3SO3::Testing

A. Marinache

Proliminarios

System

Code /isibility-Based

System Requirements-

ystem Execution

evel of Syster Detail



(Slide 37 of 41)

Application: Online Ticket Booking System

Requirements:

- Handle up to 1,000 concurrent users during peak times
- Ensure response time is under 2 seconds for any request

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code

System Requirements-

vstem Execution

evel of System



(Slide 38 of 41)

SE 3SO3::Testing

A. Marinache

Preliminaries

System State-Base

Code Visibility-Based

System

Requiremer Based

ystem Executio

evel of System

Examples

Application: Social Media App

Requirements:

- Existing features: Posting text, uploading images, liking posts
- New feature: Stories that disappear after 24 hours



(Slide 39 of 41)



SE 3SO3::Testing

A. Marinache

Preliminaries

ystem

Code Visibility-Based

System
Requirements-

System Execution

Level of System



(Slide 40 of 41)

Before we end: bit.ly/4fRCWOk

- What is the main point I learned in this set of lectures?
- What is the main point that was left out in this set of lectures?

SE 3SO3::Testing

A. Marinache

Preliminaries

System

Code Visibility-Based

System
Requirements-

ystem Execution

evel of Systemetail



William E. Perry, *Effective methods for software testing*, second edition ed., John Wiley & Sons, Inc., 2000.

SE 3SO3::Testing

A. Marinache

Preliminaries

ystem tate-Based

Code Visibility-Based

Requirements-Based

ystem Execution

evel of System

Oetail