Testing throughout the software life cycle

Software Testing: INF3121 / INF4121
Summary: Week 2

Software development **models**
- Sequential / Iterative-Incremental / Testing within a life cycle

**Test levels**
- Component (Unit) / Integration / System / Acceptance

**Test types**
- Functional / Non-functional / Software structure / Related to changes

**Maintenance testing**
Part I: Close-ended questions
Question 1

Which are **good practices for testing** within the development life cycle?

a. Early test analysis and design  
b. Different test levels are defined with specific objectives  
c. Testers will start to get involved as soon as coding is done  
d. A and B above
Which are **good practices** for **testing** within the development life cycle?

Characteristics of **good** testing

- **Early** testing
  - Every **development** activity has a corresponding **testing** activity
  - Each test **levels** has test **objectives** specific to that **level**
  - Test **analysis** + **design** should begin during the corresponding **development** activity
  - Testers should be **involved** in reviewing documents as soon as drafts are **available**
Question 1: Answer

Which are **good practices** for **testing** within the **development life cycle**?

a. Early test analysis and design
b. Different test levels are defined with specific objectives
c. Testers will start to get involved as soon as coding is done
d. A and B above
Question 2

Which option best describes objectives for test levels with a life cycle model?

a. Objectives should be generic for any test level
b. Objectives are the same for each test level
c. Objectives of a test level don’t need to be defined in advance
d. Each level has objectives specific to that level
Which option best describes **objectives for test levels with a life cycle model**?

- **System**: The behaviour of the whole system as defined by the scope of the project.
- **Unit**: Any module, program, object separately testable
- **Integration**: Interface between components; interactions with other systems (OS, HW, etc.)
- **Acceptance**: Is the responsibility of the customer - in general. The goal is to gain confidence in the system; especially in its non-functional characteristics.
Question 2: Answer

Which option best describes **objectives for test levels** with a **life cycle model**?

**Unit testing**
- Test and verify the functioning software **items**

**Integration testing**
- Test and verify the **interfaces** and **interactions** between components

**System testing**
- Test the behaviour of the **whole system** as defined by the project scope

**Acceptance testing**
- Establish **confidence** in the system | part of system | non-functional characteristics
Question 2: Answer

Which option best describes **objectives for test levels** with a **life cycle model**?

a. Objectives should be generic for any test level
b. Objectives are the same for each test level
c. Objectives of a test level don’t need to be defined in advance
d. Each level has objectives specific to that level
Question 3

Which of the following is a **non-functional quality characteristic**?

- a. Feasibility
- b. Usability
- c. Maintenance
- d. Regression
Question 3: Answer

Which of the following is a non-functional quality characteristic?

Non-functional testing

Measuring quality characteristics of software

*HOW* the system performs a certain task / function

ISO 9126 on Software Product Quality

Efficiency | Portability | Reliability | Functionality

Usability | Maintainability
Question 3: Answer

Which of the following is a non-functional quality characteristic?

a. Feasibility
b. **Usability**
c. Maintenance
d. Regression
Question 4

Which of these is a functional test?

a. Measuring response time on an on-line booking system
b. Checking the effect of high volumes of traffic in a call-centre system
c. Checking the on-line bookings screen information and the database contents against the information on the letter to the customers
d. Checking how easy the system is to use
Question 4: Answer

Which of these is a **functional test**?

**Functional testing**
- Testing *WHAT* a system should do
- Considers the *external behaviour* of the software
- Specification-based testing
  - *Black-box* testing
- Structure-based testing
  - *White-box* testing
Question 4: Answer

Which of these is a functional test?

a. Measuring response time on an on-line booking system
b. Checking the effect of high volumes of traffic in a call-centre system
c. Checking the on-line bookings screen information and the database contents against the information on the letter to the customers
d. Checking how easy the system is to use
Question 5

Which of the following is true regarding the process of fixing emergency changes?

a. There is no time to test the change before it goes live, only the best developers should do this work and should not involve testers as they slow down the process
b. Just run the retest of the defect actually fixed
c. Always run a full regression test of the whole system in case other parts of the system have been adversely affected
d. Retest the changed area and then use risk assessment to decide on a reasonable subset of the whole regression test to run in case other parts of the system have been adversely affected
Question 5: Answer

Which of the following is **true** regarding the process of fixing emergency changes?

- **Confirmation testing**
  - Retesting of software after defect has been detected and **fixed**
  - Confirm that the original **defect** has been successfully **removed**

- **Regression testing**
  - Repeated testing of already tested program after modification
  - Discover any defects introduced or uncovered as a result of change
  - Test effort **extent** is based on **risk** of finding defects in previously working software
Question 5: Answer

Which of the following is true regarding the process of fixing emergency changes?

a. There is no time to test the change before it goes live, only the best developers should do this work and should not involve testers as they slow down the process
b. Just run the retest of the defect actually fixed
c. Always run a full regression test of the whole system in case other parts of the system have been adversely affected
d. Retest the changed area and then use risk assessment to decide on a reasonable subset of the whole regression test to run in case other parts of the system have been adversely affected
Question 6

A regression test ...

a. Is only run once
b. Will always be automated
c. Will check unchanged areas of the software to see if they have been affected
d. Will check changed areas of the software to see if they have been affected
Question 6: Answer

A regression test …

Regression testing

After fixing a defect; have we introduced new defects?

Checks unchanged areas of the software

Regression test suites are run many times during development

Evolve slowly

Regression testing is a strong candidate for automation
A regression test ... 

a. Is only run once  
b. Will always be automated  
c. **Will check unchanged areas of the software to see if they have been affected**  
d. Will check changed areas of the software to see if they have been affected
Question 7

**Non-functional testing includes:**

a. Testing to see where the system does not function correctly
b. Testing the quality attributes of the system including reliability and usability
c. Gaining user approval for the system
d. Testing a system feature using only the software required for that function
Non-functional testing includes:

Non-functional testing

Testing quality characteristics → Ability testing

Aspects of software not necessarily related to specific functions

Examples

- **Reliability** testing: Fault tolerance / Robustness / Recoverability / Compliance
- **Usability** testing: Learnability / Operability / Attractiveness
- **Efficiency** testing: Time behaviour / Resource utilisation
- **Maintainability** testing: Analysability / Changeability / Stability / Testability
- **Portability** testing: Adaptability / Installability / Replaceability
Non-functional testing includes:

- Testing to see where the system does not function correctly
- Testing the quality attributes of the system including reliability and usability
- Gaining user approval for the system
- Testing a system feature using only the software require for that function
_____ testing is performed by customers at their own site
Question 8: Answer

_____ testing is performed by customers at their own site

Acceptance testing

Responsible: Customers / Users of the system

Alpha testing

Performed at the site of the developing organisation

Beta testing

Field testing

Performed by people at their own locations

User acceptance / Operational / Contract and regulation testing
Question 8: Answer

_____ testing is performed by customers at their own site

BETA testing
## Question 9

### Pair the following test levels with their description

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Unit level</td>
<td>A. Tests the behavior of the whole system</td>
<td></td>
</tr>
<tr>
<td>2. Integration level</td>
<td>B. Performed by customers</td>
<td></td>
</tr>
<tr>
<td>3. System level</td>
<td>C. Tests any module or object separately testable</td>
<td></td>
</tr>
<tr>
<td>4. Acceptance level</td>
<td>D. Tests the interactions of the interfaces of the system</td>
<td></td>
</tr>
</tbody>
</table>
**Question 9: Answer**

Pair the following **test levels** with their **description**

<table>
<thead>
<tr>
<th>1. Unit level</th>
<th>A. Tests the <strong>behavior</strong> of the <strong>whole system</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Integration level</td>
<td>B. Performed by <strong>customers</strong></td>
</tr>
<tr>
<td>3. System level</td>
<td>C. Tests any <strong>module</strong> or <strong>object separately</strong> testable</td>
</tr>
<tr>
<td>4. Acceptance level</td>
<td>D. Tests the <strong>interactions</strong> of the <strong>interfaces</strong> of the <strong>system</strong></td>
</tr>
</tbody>
</table>
Acceptance testing is not the responsibility of the development team. It is the responsibility of the customers, but the development team can assist in the process.

a. True
b. False
Acceptance testing is not the responsibility of the development team. It is the responsibility of the customers, but the development team can assist in the process.

Acceptance testing

- Establish confidence in the system | part of system
- Examines non-functional characteristics of the system
- Verify the fitness for use
- Responsibility lies with the customers / users
- Stakeholders may be involved in the testing process
Acceptance testing is not the responsibility of the development team. It is the responsibility of the customers, but the development team can assist in the process.

a. True
b. False
Part II: Exercises and Open-ended questions
Exercise: Different Types of Testing

Go to www.ikea.com/no/no/

Give examples of possible …

a. Unit tests
b. Integration tests
c. System tests
Unit Tests

Choosing a department

Choosing furniture
Integration Tests

Setting a filter

Price level / Colour
System Tests

Choose an item

Place into shopping cart
Why do you think we need to test at integration level top-down or bottom-up rather than big-bang?

Why is acceptance testing important?

Why do you think it is important to test on-site?
Top-down Testing

**Top-down testing**

- Testing from *top* to *bottom*
- From *main* module to *sub-modules*
- Follows *control flow* or architectural structure

If *lower-level* modules have *not* yet been *developed*

- *Simulate* the *lower-level* *modules* using STUBS
- *Temporary programs* substitute components / systems
Bottom-up testing

Testing from bottom of the control flow upwards

From lower-level modules to higher-level modules

Each lower hierarchy component is tested

If higher-level modules have not yet been developed

Simulate the higher-level modules using DRIVERS

Temporary programs to substitute main components or higher-level systems
Importance of Acceptance Testing

Acceptance testing

Does the software product meet the customer’s needs?

Does the software product comply with the specified requirements / documentation?

Last chance of the customer to discard an insufficient / inadequate product

Adequate testing protects customer from losses caused by poor software products

Compares results to expectations

Validation

Did we build the *right thing*?

Verification

Did we build the *thing right*?
Importance of Acceptance Testing

- How the Customer explained it
- What the Project Manager understood
- How the Analyst designed it
- What the Programmer wrote
- What the Business Consultant presented
- How the Project was documented
- What Operations installed
- How the Customer was billed
- How the Solution was supported
- What the Customer really needed

https://i.stack.imgur.com/2YGv6.png
Importance of on-Site Testing

On-site testing

- Get application into the hands of the customers.
  - The people who will actually be using the software product.

- Provides a “real”, overall experience of the software.
  - Realistic environment to mimic actual use after deployment.

- Potentially more honest user feedback.

Other benefits:

- Improve software quality.
- Bug detection.
- Cost reduction.
Assignments
2-4 people in each group
Alt. I: Register as an individual. We form the groups
Alt. II: Register the entire group at once.

Next week:
Static test techniques
Yulai Fjeld

Master student

Department of Informatics
University of Oslo

Previously taught courses

Systemutvikling (INF1050), Universitet i Oslo
Software Testing (INF3121/4121), Universitetet i Oslo
Systemutvikling (ADSE2200), Høgskolen i Oslo og Akershus