Software Testing

University of Oslo, **Department of informatics**
Spring 2018

Agenda

Chapter 1
The fundamentals of testing

Introduction

Course Structure

Recap of the first week

Weekly exercises

Summary: Week 1

Why is testing necessary?

What is testing?

Test principles

Seven test principles in total

Fundamental test process

Five important activities in total

Psychology of testing

Part I: Close-ended questions

Question 1

A company recently purchased a commercial off-the-shelf application to automate their bill-paying process. They now plan to run an acceptance test against the package prior to putting it into production. Which is their most likely reason for testing?

a. To build confidence in the application
b. To detect bugs in the application
c. To gather evidence for a lawsuit
d. To train the users

Question 1: Answer

Which is their most likely reason for testing?

Characteristics of acceptance testing

Establish confidence the system | part of system

Examine non-functional characteristics of the system

Verify the fitness for use

Responsibility lies with the customers / users

Different types of acceptance testing

User acceptance testing

Operational testing

Contract and regulation acceptance testing

Question 1: Answer

A company recently purchased a commercial **off-the-shelf** application to automate their bill-paying process. They now plan to run an **acceptance test** against the package **prior** to putting it into **production**. Which is their most likely **reason** for **testing**?

a. **To build confidence in the application**
b. To detect bugs in the application
c. To gather evidence for a lawsuit
d. To train the users

Question 2

According to the ISTQB Glossary, the word 'bug' is synonymous with which of the following words?
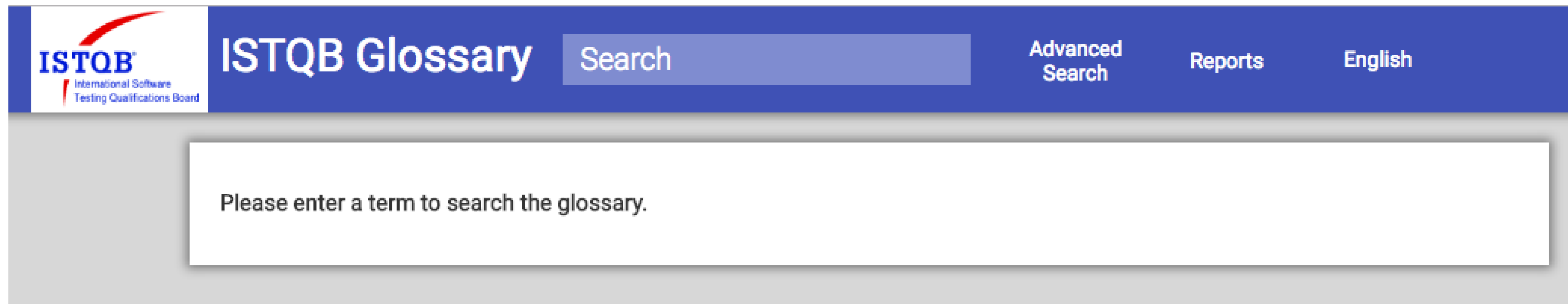
a. Incident
b. Defect
c. Mistake
d. Error

Question 2: Answer

According to the ISTQB Glossary, the word 'bug' is synonymous with which of the following words?

ISTQB Glossary

www.astqb.org/**glossary**/

Question 2: Answer

According to the ISTQB Glossary, the word 'bug' is synonymous with which of the following words?

Incident

"*Any event occurring that requires investigation*"

Defect

"*Flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition.*"

Mistake / Error

"*A human action that produces an incorrect result*"

Question 2: Answer

According to the ISTQB Glossary, the word 'bug' is synonymous with which of the following words?

a. Incident
b. **Defect**
c. Mistake
d. Error

Question 3

According to the **ISTQB Glossary**, a _____ relates to **negative** consequences that **could occur**.

Question 3: Answer

According to the ISTQB Glossary, a _____ relates to negative consequences that could occur.

What could go wrong?

Estimation and scheduling

Sudden changes to requirements

Employees leaving, causing delays

Developers showing off skills, adding unnecessary features

These factors could result in *"future negative consequences"*

How to handle them? → Risk management

Question 3: Answer

According to the **ISTQB Glossary**, a _____ relates to **negative** consequences that **could occur**.

RISK

Question 4

Ensuring that test design starts during the requirements definition phase is important to enable which of the following test objectives?

a. Preventing defects in the system
b. Finding defects through dynamic testing
c. Gaining confidence in the system
d. Finishing the project on time

Question 4: Answer

Why is it important to start test design during the requirements definition phase?

What is the purpose of testing?

Finding defects

Reduce probability of undiscovered defects

Preventing defects

Ensure, as far as possible, that defects are not introduced

Gaining confidence in the level of quality

Presence / Absence of defects to indicate SW system quality

Provide information for decisions-making

Testing grants different *insight* into various aspects of software

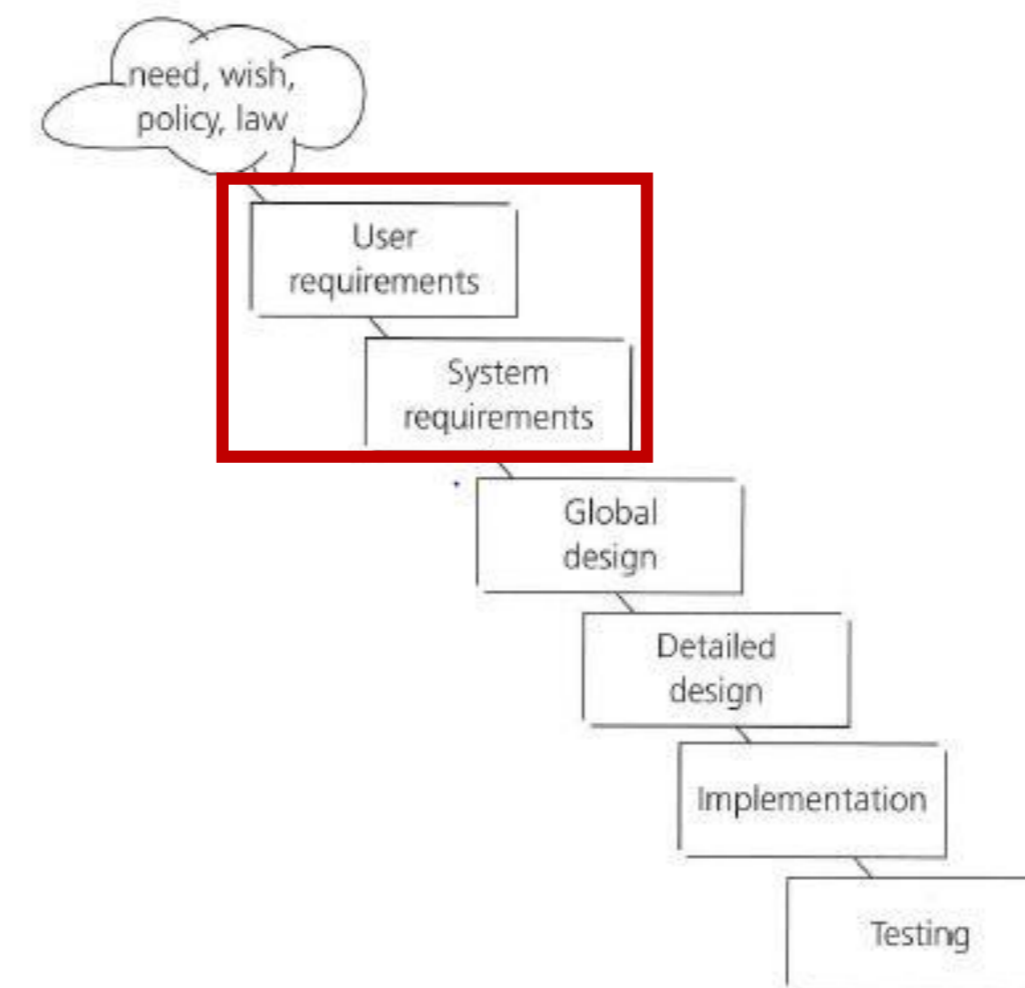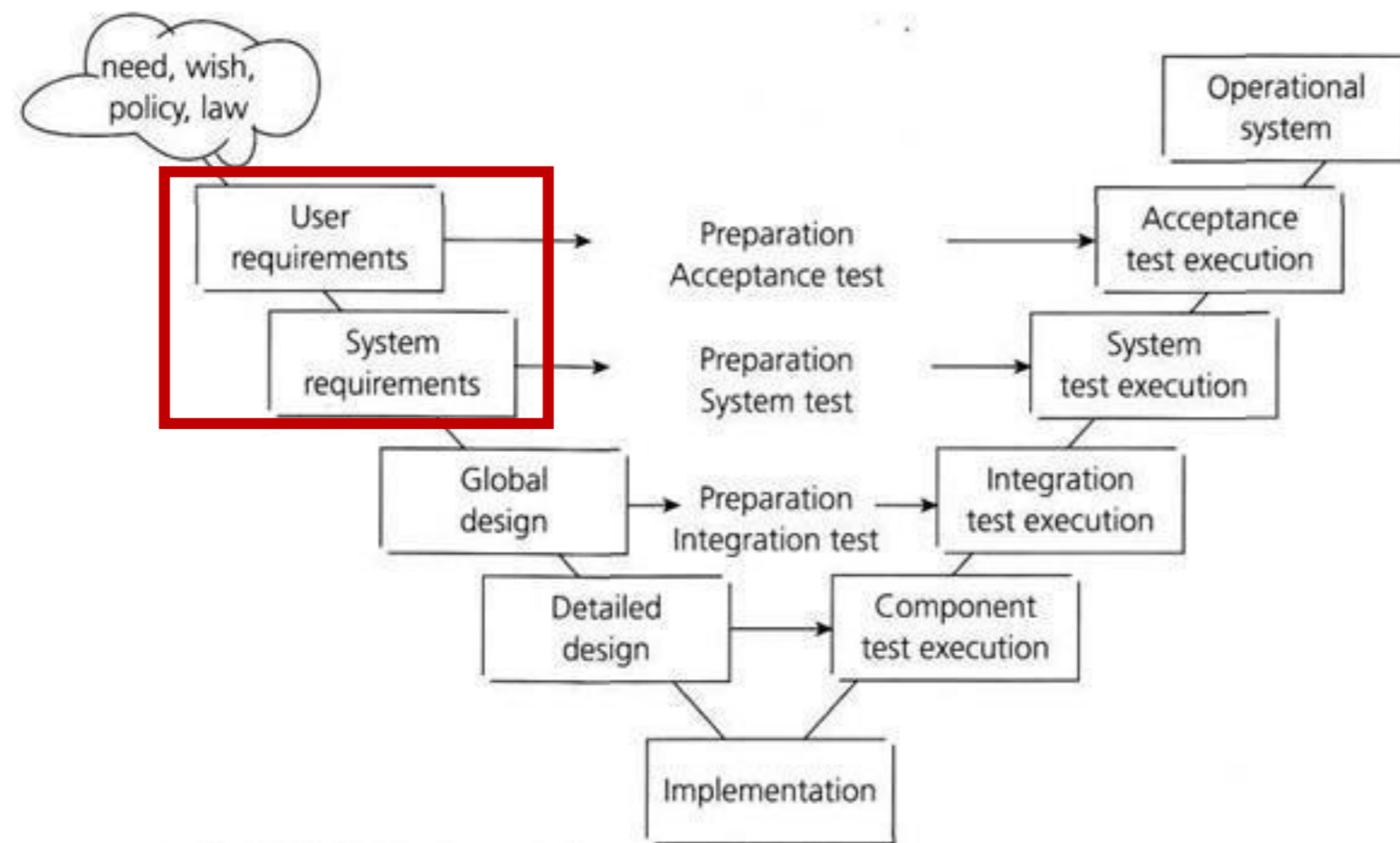# Question 4: Answer

Why is it important to start test design during the requirements definition phase?

Why start during the requirements definition phase?

Early testing

Testing should start as soon as possible

Question 4: Answer

Ensuring that **test design** starts during the **requirements** definition **phase** is **important** to enable which of the following test **objectives**?

a. **Preventing defects in the system**
b. Finding defects through dynamic testing
c. Gaining confidence in the system
d. Finishing the project on time

Question 5

An **exhaustive test suite** would include:

a. All combinations of input values and preconditions
b. All combinations of input values and output values
c. All pairs of input values and preconditions
d. All states and state transitions

Question 5: Answer

An exhaustive test suite would include:

Test suite

Logical collection of several test cases

Test software program to show specified set of behaviours

Question 5: Answer

An **exhaustive test suite** would include:

a. **All combinations of input values and preconditions**
b. All combinations of input values and output values
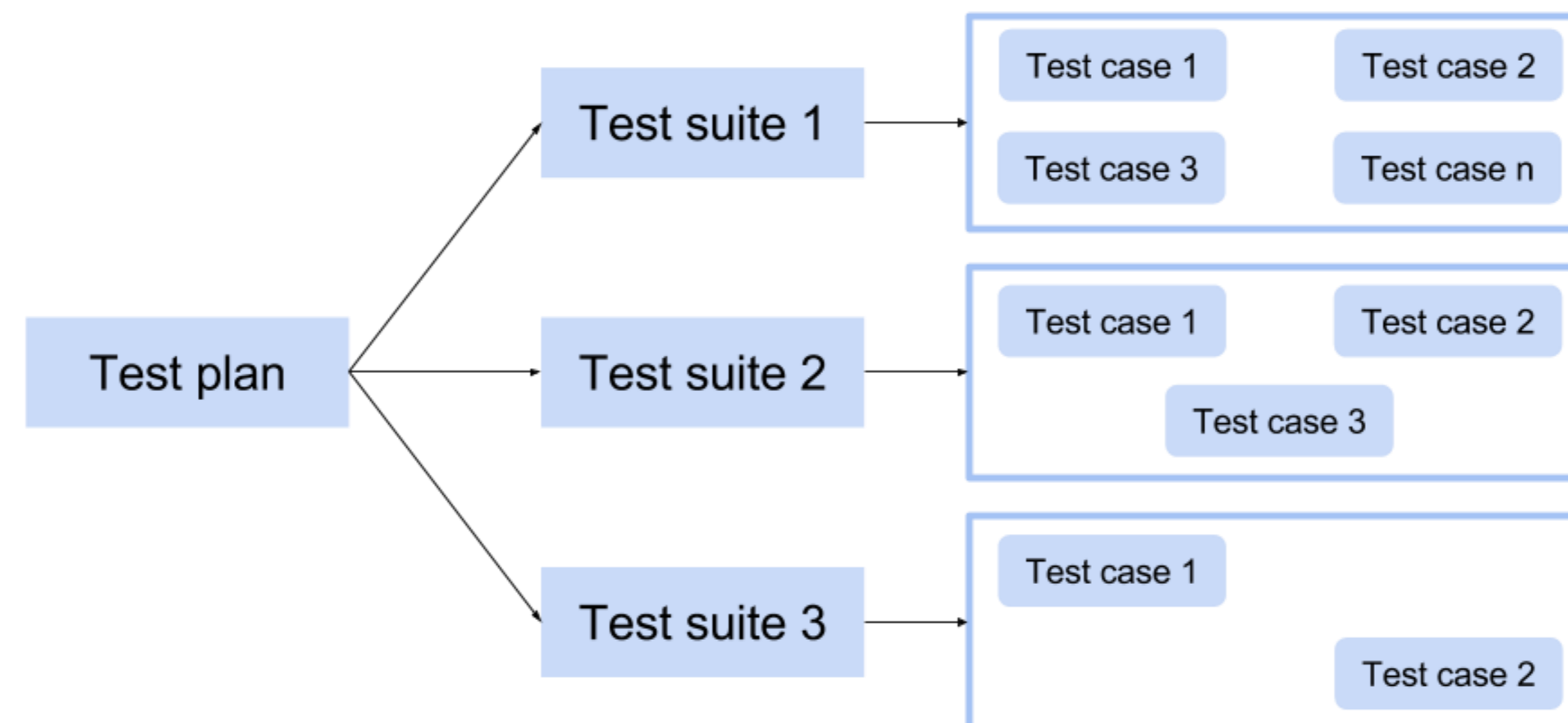c. All pairs of input values and preconditions
d. All states and state transitions

Question 6

According to the ISTQB Glossary, regression testing is required for what purpose?

a. To verify the success of corrective actions
b. To prevent a task from being incorrectly considered completed
c. To ensure that defects have not been introduced by a modification
d. To motivate better unit testing by the programmers

Question 6: Answer

According to the ISTQB Glossary, regression testing is required for what purpose?

Regression testing

After fixing a defect; have we introduced new defects?

Checks unchanged areas of the software

Confirmation testing

Retesting of software after defect has been detected and fixed

Confirm that the original defect has been successfully removed

Question 6: Answer

According to the ISTQB Glossary, **regression testing** is required for what **purpose**?

a. To verify the success of corrective actions
b. To prevent a task from being incorrectly considered completed
c. **To ensure that defects have not been introduced by a modification**
d. To motivate better unit testing by the programmers

Question 7

Which of the following is most **important** to promote and maintain **good relationships** between **testers** and **developers**?

a. Understanding what managers value about testing
b. Explaining test results in a neutral fashion
c. Identifying potential customer work-arounds for bugs
d. Promoting better quality software whenever possible

Question 7: Answer

Which of the following is most important to promote and maintain good relationships between testers and developers?

Importance of good team dynamics

People execute the process → People affect the result

Helping developers improve → Better relationship to testers

Promote and nurture trusting inter-team relationships

Constructive feedback

Learning opportunities

Question 7: Answer

Which of the following is most **important** to promote and maintain **good relationships** between **testers** and **developers**?

a. Understanding what managers value about testing
b. **Explaining test results in a neutral fashion**
c. Identifying potential customer work-arounds for bugs
d. Promoting better quality software whenever possible

Question 8

Which of the statements below is the best assessment of how test principles apply across the test life-cycle?

a. Test principles only affect the preparation for testing
b. Test principles only affect test execution activities
c. Test principles affect the early test activities such as review
d. Test principles affect activities throughout the test life-cycle

Question 8: Answer

Which of the statements below is the best assessment of how **test principles** apply across the **test life-cycle**?

P1: Testing shows presence of defects
- Testing can show that defects are present, but cannot prove there are no defects.
- Testing reduces the probability of undiscovered defects remaining in the software; but even if no defects are found, this is not a proof of correctness.

P2: Exhaustive testing is impossible
- Testing everything is not feasible. We use risks and priorities to focus on test effort.

P3: Early testing
- Testing should start as soon as possible in the development life-cycle and should be focused on defined objectives.

P4: Defect clustering
- A small number of modules contain most of the defects discovered during pre-release testing.

P5: Pesticide paradox
- If the same set of tests will be repeated over and over, it will no longer find new bugs.

P6: Testing is context dependent
- I.e. safety-critical SW is tested differently from an e-commerce site.

P7: Absence-of-errors fallacy
- Finding and fixing defects does not help if the SW system is unstable or does not meet the user's expectations.

Question 8: Answer

Which of the statements below is the best assessment of how test principles apply across the test life-cycle?

Test principles deal with

Good practices for software testing

Early testing

Importance of testing throughout the test life-cycle

Limitations of testing

Exhaustive testing / Presence of defects

Considerations and challenges of software testing

Question 8: Answer

Which of the statements below is the best assessment of how **test principles** apply across the **test life-cycle**?

a. Test principles only affect the preparation for testing
b. Test principles only affect test execution activities
c. Test principles affect the early test activities such as review
d. **Test principles affect activities throughout the test life-cycle**

Is "**evaluation** of **exit criteria**" for of the **fundamental test process**?

a. True
b. False

Is "**evaluation of exit criteria**" for of the **fundamental test process**?

Fundamental test process

Testing is *not* a single activity

Activities for each step in the test process → From planning to test closure

Activities

1   Plan and Control
2   Analysis and Design
3   Implementation and Execution
4   Evaluate Exit Criteria and Report
5   Test Closure Activities

Question 9: Answer

Is "**evaluation** of **exit criteria**" for of the **fundamental test process**?

1. Plan and Control

What / How / When / By whom?

2. Analysis and Design

Review test basis / Analyse test objectives

Design test cases, environments, data

3. Implementation and Execution

Group, prioritise test scripts / Write automated test scenarios

Run tests / Compare results / Repeat / Log outcome

Is "**evaluation of exit criteria**" for of the **fundamental test process**?

4. Evaluate Exit Criteria and Report

Assess test execution against defined objectives

Definition of Done (DoD)

Write / Extract test summary / Produce report for stakeholders

5. Test Closure Activities

Assess test effort: What has been delivered?

Archive delivered items (SW, tests, documentation) / Analyse lessons learned

Question 9: Answer

Is "**evaluation** of **exit criteria**" for of the **fundamental test process**?

a. **True**
b. False

Pair the following processes with their corresponding activities:

1. Test plan	A. Group tests into scripts
2. Test analysis	B. Write or extract a test summary report for the stakeholders
3. Test implementation	C. Transform test objectives into test conditions and test cases
4. Test reporting	D. Establish the scope, objectives, and risks of testing

Question 10: Answer

Pair the following **processes** with their corresponding **activities**:

1. Test **plan**	A. **Group** tests into **scripts**
2. Test **analysis**	B. Write or **extract** a test **summary report** for the stakeholders
3. Test **implementation**	C. **Transform** test **objectives** into test **conditions** and test **cases**
4. Test **reporting**	D. Establish the **scope**, **objectives**, and **risks** of testing

Part II: Exercises and Open-ended questions

Exercise

A test team consistently **finds** between **90** and **95%** of the **defects present** in the **system** under test. While the **test manager** understands that this is a **good defect-detection** percentage, **senior management** and executives remain **disappointed**, saying that the test team **misses too many bugs**. Given that the **users** are generally **happy** with the system and that the **failures** which have occurred have generally been **low impact**, which of the **testing principles** is most likely to help the test manager **explain** why some **defects** are **likely** to be **missed**?

Exhaustive testing is impossible
Defect clustering
Pesticide paradox
Absence-of-errors fallacy
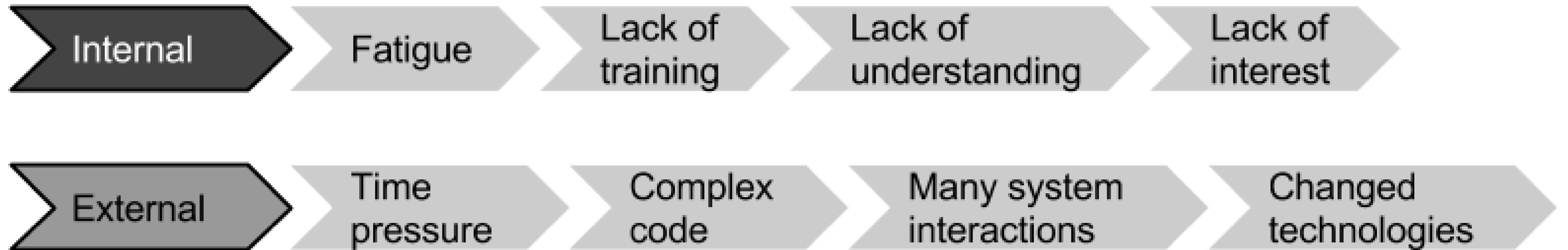
Open-Ended Questions

Can you give **examples** of **internal** and **external causes** of **defects**? Which one do you consider to be the most **important**?

Why is it important to **check** a software **system** if it is **fit for use**?

What **qualities** should a **tester** have, if you were to **hire** him?

Internal and External Causes

Internal → Fatigue → Lack of training → Lack of understanding → Lack of interest

External → Time pressure → Complex code → Many system interactions → Changed technologies

Checking if SW is Fit for Use

Fit for use → 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*?

Qualities of a Tester

Problem-oriented

Desire to find problems / Ability to solve problems
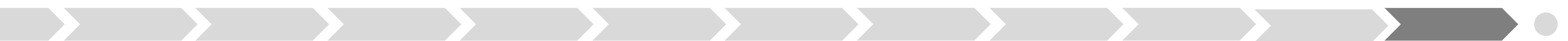
Detail-oriented

Curiosity and Creativity

Approaching the test effort from new, varied angles

Professional pessimism

Being realistic about the product / human behavioural patterns

Good interpersonal and communication skills

Courtesy, understanding of others, good attitude, diplomatic, etc.

The seminar slides are made by

Yulai Fjeld   **ydfjeld @ uio.no**

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