

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 1 : Introduction



Spring 2024

By Marwa Solla

What We Learn In This Lecture

What is Software?

What is a Quality?

What is Software quality?

The Concept of Software Quality:

- Quality Assurance (QA),
- Quality Control (QC)
- Testing

What is a Software?

- Let us review the IEEE definition for “software” (IEEE, 1991), shown in Frame 2.1.

Frame 2.1 **Software – IEEE definition**

Software is:

Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

- The IEEE definition of software, lists the following four components of software:
 - Computer programs (the “code”)
 - Procedures
 - Documentation
 - Data necessary for operating the software system.

All four components are needed in order to assure the quality of the software development process and the coming years of maintenance services.

What is a quality?

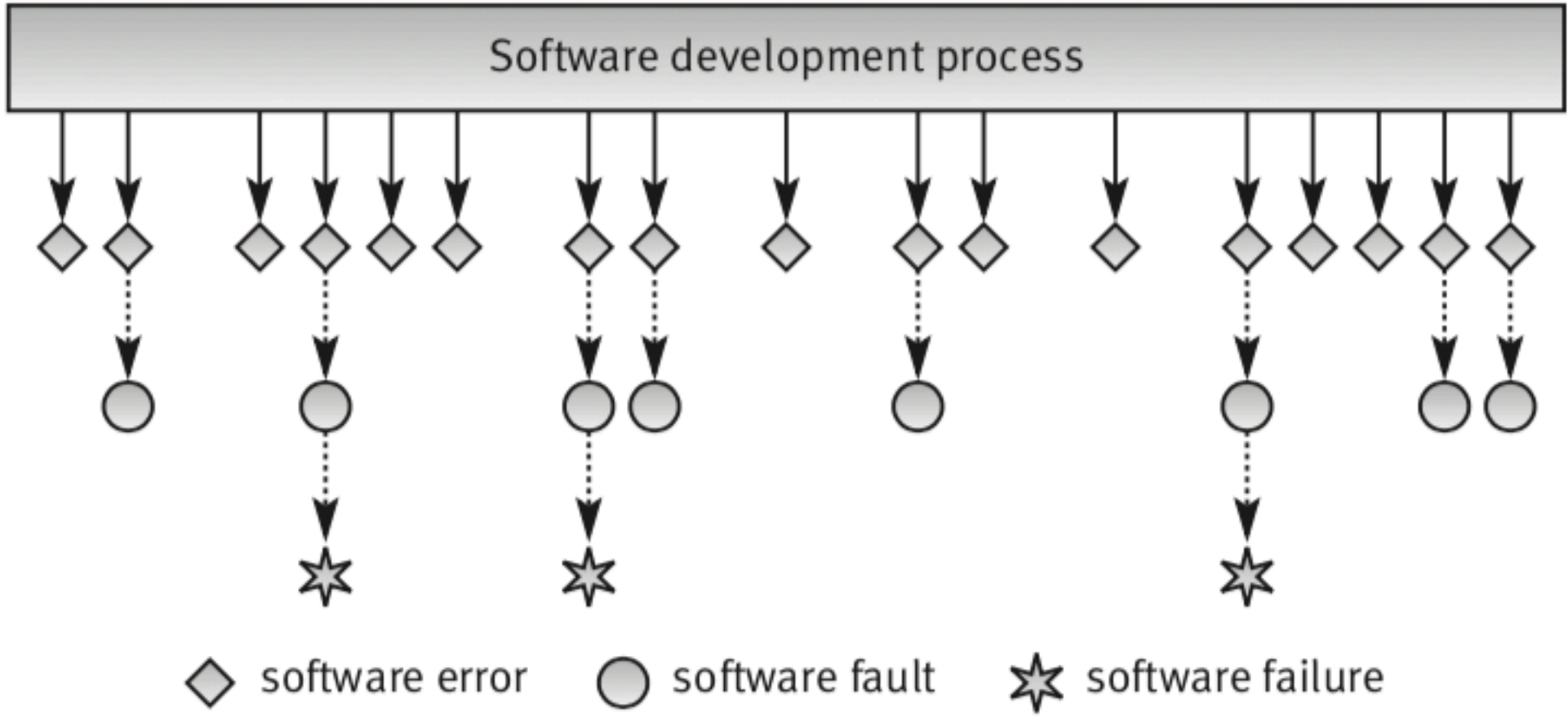
- A definition of quality is "your focus on the satisfying your customer's needs".
- Customer satisfaction: The only true measure of acceptable quality is customer satisfaction, which takes into account both interpretations of the needs and expectations of customers.

What is a quality?

Quality has many meanings:

- Conformance with requirements,
- The totality of characteristics of an entity that bear on its ability to satisfy stated or implied needs,
- Fitness for use,
- Freedom from defects,
- Delighting customers.

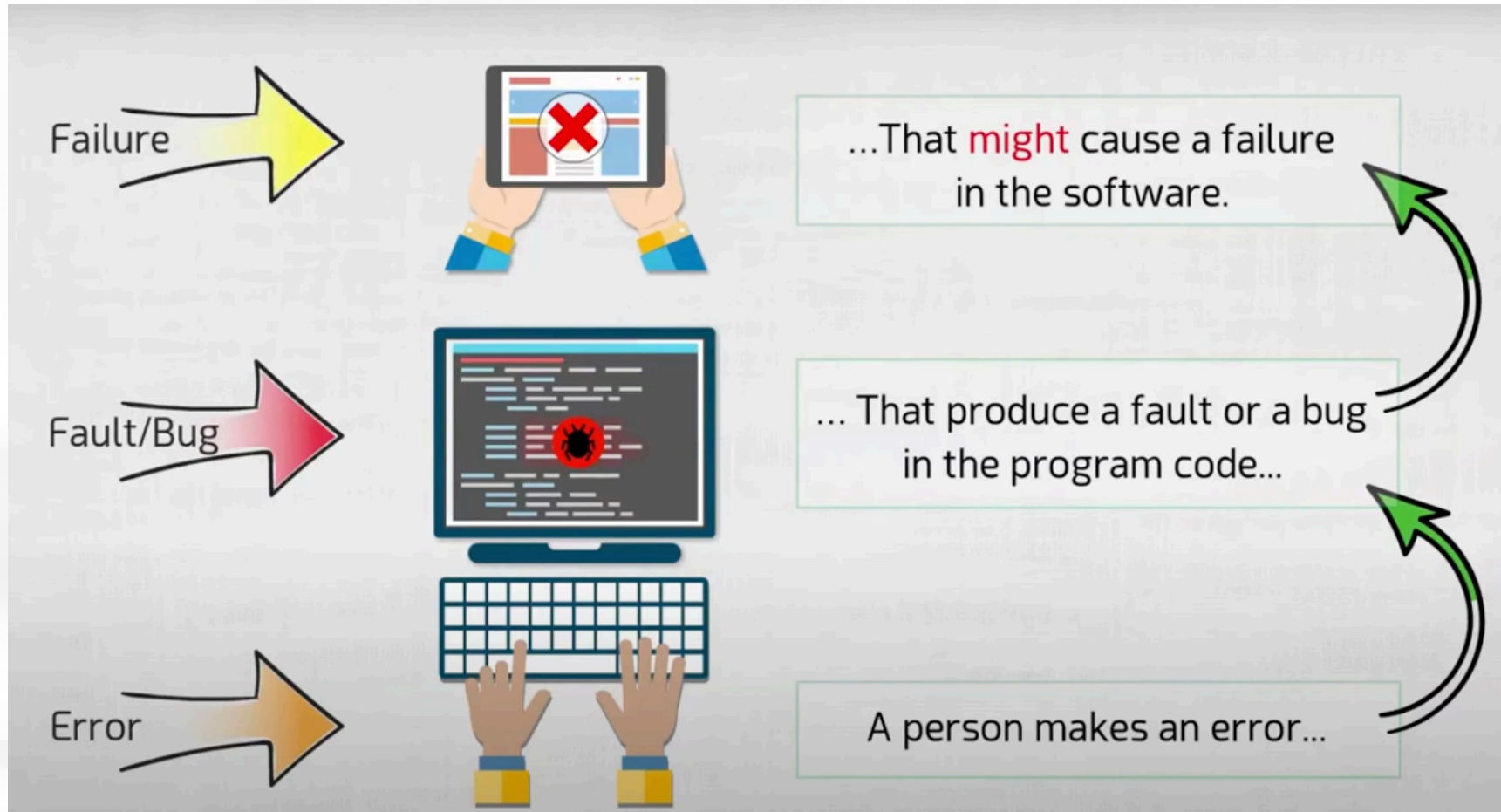
Software Errors, Faults and Failures



Software Errors, Faults and Failures

- ❑ Software errors are sections of the code that are partially or totally incorrect as a result of a grammatical, logical or other mistake made by a systems analyst, a programmer, or another member of the software development team.
- ❑ Software faults are software errors that cause the incorrect functioning of the software during a specific application.
- ❑ Software faults become software failures only when they are “*activated*”, that is, when a user tries to apply the specific software section that is faulty. Thus, the root of any software failure is a software error.

Software Errors, Faults and Failures



Software Errors, Faults and Failures

- **Types of errors**
 1. code error
 2. procedure error
 3. documentation error
 4. software data error

Classification of the causes of software Errors

As software errors are the cause of poor software quality, it is important to investigate the causes of these errors in order to prevent them. Frame 2.2 summarizes the causes of software errors.

Frame 2.2 **The nine causes of software errors**

1. Faulty requirements definition
2. Client–developer communication failures
3. Deliberate deviations from software requirements
4. Logical design errors
5. Coding errors
6. Non-compliance with documentation and coding instructions
7. Shortcomings of the testing process
8. Procedure errors
9. Documentation errors

Classification of the causes of software Errors

1. Faulty requirements definition

- Erroneous definition of requirements.
- Absence of vital requirements.
- Incomplete definition of requirements.
- Inclusion of unnecessary requirements, functions that are not expected to be needed in the near future.

Classification of the causes of software Errors

2. Client–developer communication failures

- Misunderstanding of the client's instructions as stated in the requirement document.
- Misunderstanding of the client's requirements changes presented to the developer in written form during the development period.
- Misunderstanding of the client's requirements changes presented orally to the developer during the development period.
- Misunderstanding of the client's responses to the design problems presented by the developer.
- Lack of attention to client messages referring to requirements changes and to client responses to questions raised by the developer on the part of the developer.

Classification of the causes of software Errors

3. Deliberate deviations from software requirements

- The developer reuses software modules taken from an earlier project without sufficient analysis of the changes and adaptations needed to correctly fulfill all the new requirements.
- Due to time or budget pressures, the developer decides to omit part of the required functions in an attempt to cope with these pressures.

Classification of the causes of software Errors

4. logical design errors

- Definitions that represent software requirements by means of erroneous algorithms.
- Process definitions that contain sequencing errors.
- Erroneous definition of boundary conditions.

Classification of the causes of software Errors

5. coding errors

6. non-compliance with documentation and coding instructions

7. shortcomings of the testing process

- Incomplete test plans leave untreated portions of the software or the application functions.
- Failures to document and report detected errors and faults.
- Incomplete correction of detected errors due to time pressures.

Classification of the causes of software Errors

8. Procedure errors

- Describing an incorrect procedure.

9. Documentation errors

- Omission of software functions.
- Errors in the instructions given to users, resulting in incorrect applications.
- Listing of non-existing software functions, that is, functions planned in the early stages of development but later dropped, and functions that were active in previous versions of the software but cancelled in the current version.

What is a software quality?

The definition suggested by IEEE (IEEE, 1991) shown in Frame 2.3 is the definition we have chosen to apply in this text.

Frame 2.3 Software quality – IEEE definition

Software quality is:

1. The degree to which a system, component, or process meets specified requirements.
2. The degree to which a system, component, or process meets customer or user needs or expectations.

هي مدى تطابق مكونات او اجراءات نظام ما للمطلبات المحددة.

هي مدى تلبية مكونات او اجراءات النظام لحاجة وتوقعات المستخدم.

What is a software quality?

Additional aspects of software quality are included in the definition suggested by Pressman

Frame 2.4 **Software quality – Pressman’s definition**

Software quality is defined as:

Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

“مدى الالتزام بالمتطلبات الواضحة المحددة و بمعايير تطوير موثقة وكذلك الخصائص الضمنية للمتطلبات المتوقعة في المنتج البرمجي

Software Quality

Quality: Quality is defined as justification of all the requirements of a customer in a product.

Quality software is reasonably

- Bug-free.
- Delivered on time.
- Within budget.
- Meets requirements and/or expectations.
- Maintainable.

What is Quality Assurance (QA)?

One of the most commonly used definitions of software quality assurance (SQA) is offered by the IEEE Glossary (IEEE, 1991), cited in Frame 2.5.

Frame 2.5 **Software quality assurance – The IEEE definition**

Software quality assurance is:

1. A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
2. A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with quality control.

What is Quality Assurance (QA)?

A part of quality management focused on providing confidence that quality requirements will be fulfilled”.

It is a broad term, explained on the Google Testing Blog as “the continuous and consistent improvement and maintenance of process that enables the QC job”. As follows from the definition, QA focuses more on organizational aspects of the quality management, monitoring the consistency of the production process.

What is Quality Control (QC)?

- A part of quality management focused on fulfilling quality requirements.
- Quality control is a set of activities carried out with the main objective of withholding products from shipment if they do not qualify

What is Software Testing?

- Software Testing is a part of software development process.
- Software Testing is an activity to detect and identify the defects in the software.
- The objective of testing is to release quality product to the client.

What is Software Testing?

- Testing is the basic activity aimed at detecting and solving technical issues in the software source code and assessing the overall product usability, performance, security and compatibility.
- It has a very narrow focus and is performed by the test engineers in parallel with the development process or at the dedicated testing stage (depending on the methodology approach to the software development cycle).

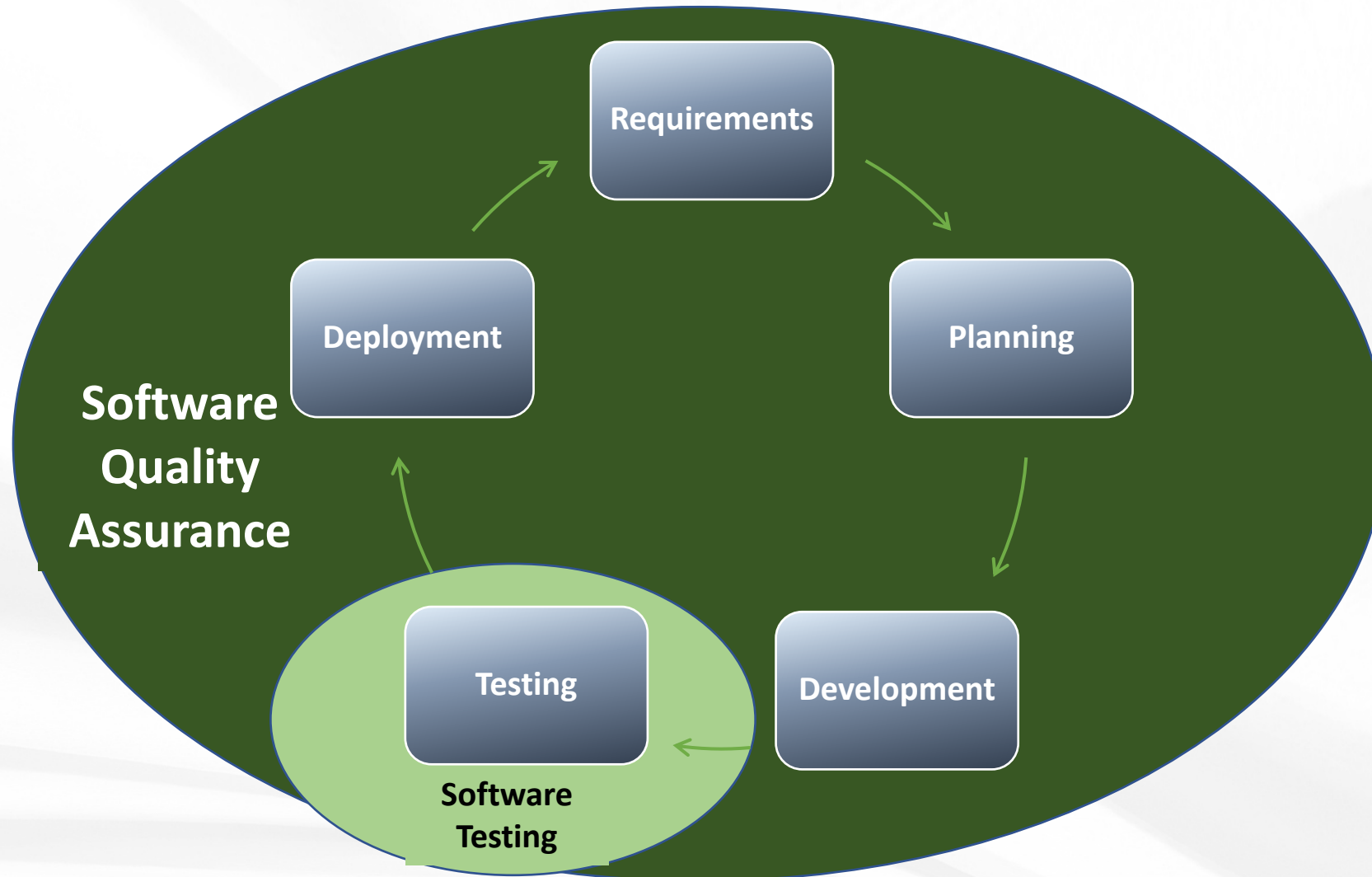
Difference between QA , QC, and Software Testing?

| | QA | QC | Testing |
|----------------|--|--|---|
| Purpose | Setting up adequate processes, introducing the standards of quality to prevent the errors and flaws in the product | Making sure that the product corresponds to the requirements and specs before it is released | Detecting and solving software errors and flaws |
| Focus | Processes | Product as a whole | Source code and design |
| What | Prevention | Verification | Detection |
| Who | The team including the stakeholders | The team | Test Engineers, Developers |

Difference between QA , QC, and Software Testing?

| | QA | QC | Testing |
|---------------------------|---|---------------------------------|--|
| When | Throughout the process | Before the release | At the testing stage or along with the development process |
| Starting point | Project planning stage (most efficient) Any point in the SDLC when the project is in trouble | Requirement gathering stage | Testing stage |
| Tools and measures | Quality metrics Reviews Audits | Testing metrics Test reports | Types of testing |

Difference between QA , QC, and Software Testing?



Explain The objectives of SQA activities

The objectives of SQA activities for software development and maintenance are:

- 1) Obtain an acceptable level of confidence that the product will meet specified functional requirements
- 2) Ensuring an acceptable level of confidence in delivering the product on time and within the specified budget
- 3) managing activities for the improvement and greater efficiency of software development and SQA activities.

In sum:

- (1) Quality control and quality assurance activities serve different objectives.
- (2) Quality control activities are only a part of the total range of quality assurance activities.

The end