



Fundamentals of Software Testing

WS2019

Markus Zoffi



peso@inso.tuwien.ac.at
<https://peso.inso.tuwien.ac.at>



INSO - Industrial Software

Institut für Information Systems Engineering | Fakultät für Informatik | Technische Universität Wien

Content

I Fundamentals

1 Importance of Testing

2 Definitions

3 ISTQB Testing principles

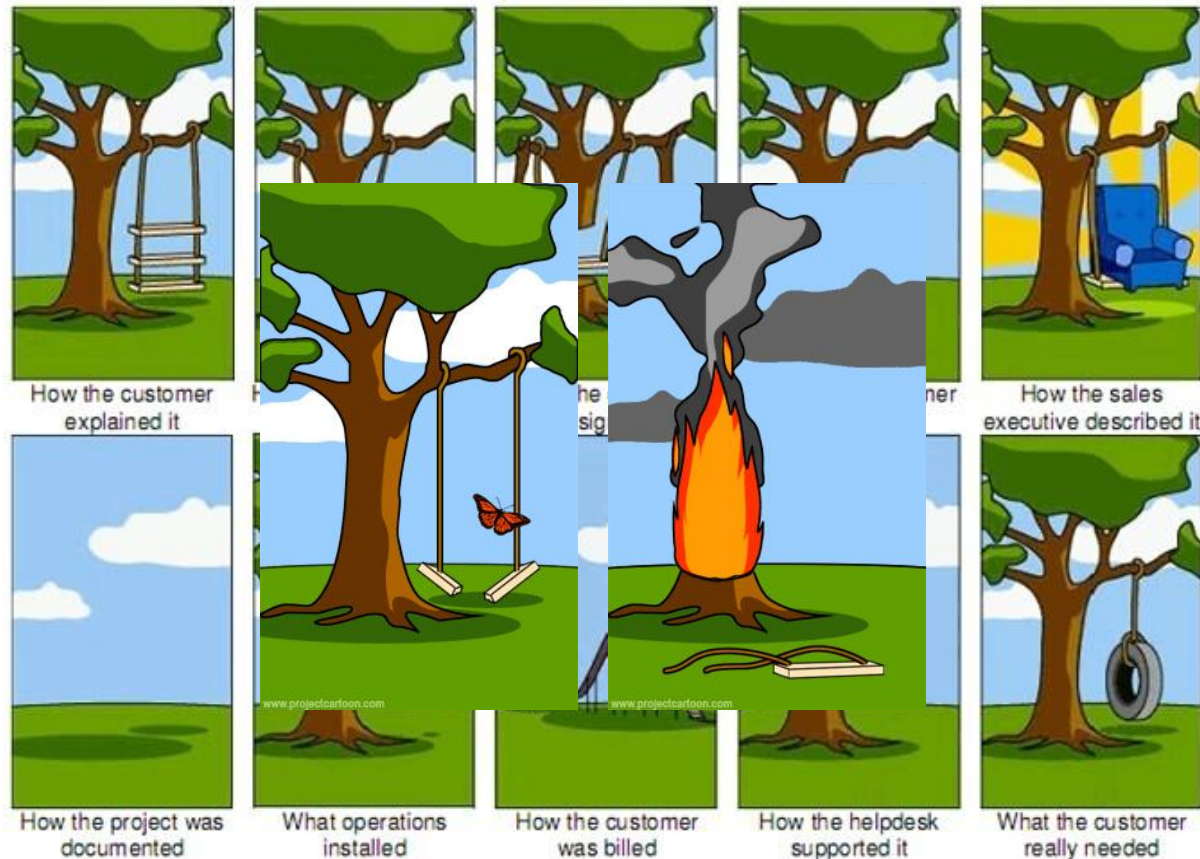
4 ISTQB Fundamental Test Process

5 Conclusion

Importance of Testing

Why do we need testing anyway?

What could possibly go wrong?



Examples of Software Failures

- **1985 – 1987 Therac-25 - radiation therapy machine**
 - Previous models had hardware interlocks
 - Therac-25 replaced them by software interlocks
 - The software interlock could fail due to a race condition
- a one-byte counter frequently overflowed
- manual input at a precise moment caused the overflow
- struck the patients with approximately 100 times the intended dose of radiation
- radiation poisoning
- some patients later died as a result of the overdose

Examples of Software Failures

- **1986 - Inversion when flying over the equator**
 - Programm designed for Missile-Navigation
 - When crossing the equator use negative signs for coordinations
 - Results in an 180° rotation of the missile
 - Slightly adopted code was used for F16 Jets
- In a simulation - flying the F-16 over the equator results in a instantly flipped over plane

Examples of Software Failures

- **1996 - Explosion of the Ariane 5**
 - The unmanned Ariane 5 rocket exploded just 40 seconds after lift-off
 - Explosion was the result of a software error:
 - Unhandled exception after a 64 bit integer was converted to a 16 bit signed integer
 - Software module was reused without proper testing from Ariane 4 rocket.
 - The error was not supposed to happen with Ariane 4
 - The destroyed rocket and its cargo were valued at \$500 million



Examples of Software Failures

- **2014 - Amazon - glitch cuts prices to 1p**
 - Third-party sellers can use automatic price calculation software
 - The lowest possible product price, considering a competitors price as well as economical profit would be determined
- The security measurements failed and reduced prices to as little as 1p

Examples of Software Failures

- **2019 – Airbus – software bug forces airline to turn off and on airplanes every 149 hours**
 - A software bug that's related to the plane's internal timer could potentially trigger some concerning side effects (unexplained failures of potentially flight-critical digital systems) if not properly reset and resolved with a hard reboot.
 - An update would require the plane to be taken out of service (incl. maintenance, quality assurance testing, ...) and would take longer to perform than a simple power cycle.
 - Workaround: reboot the aircraft every 149 hours.

Examples of Software Failures

- **2019/2020 – Boeing**
- **Severe Software Bug**
- **Caused Plane crashes**
 - grounding of Boeing 737 Max
 - Loss in reputation, trust, money, ...
- "This airplane is designed by clowns who in turn are supervised by monkeys," one employee wrote.

Examples of Software Failures

- **Many more software Failures**
 - 1983 – Soviet nuclear false alarm incident
 - 2000 – Year 2000 Bug
 - 2004 – Telekom Shutdown(Vienna/Lower Austria)
 - 2010 – Skype Downtime
 - 2014 – Heartbleed
 - 2017 – Instagram exploit
 - 2018 – Autonomous Car Crash
 - 2038 – Year 2038 Bug?
 - Many computers handle dates by counting seconds since 01/01/1970 (Unix time)
 - stored on as a signed 32-bit integer (max: 2.147.483.647)
 - 19. Januar 2038 03:14:08

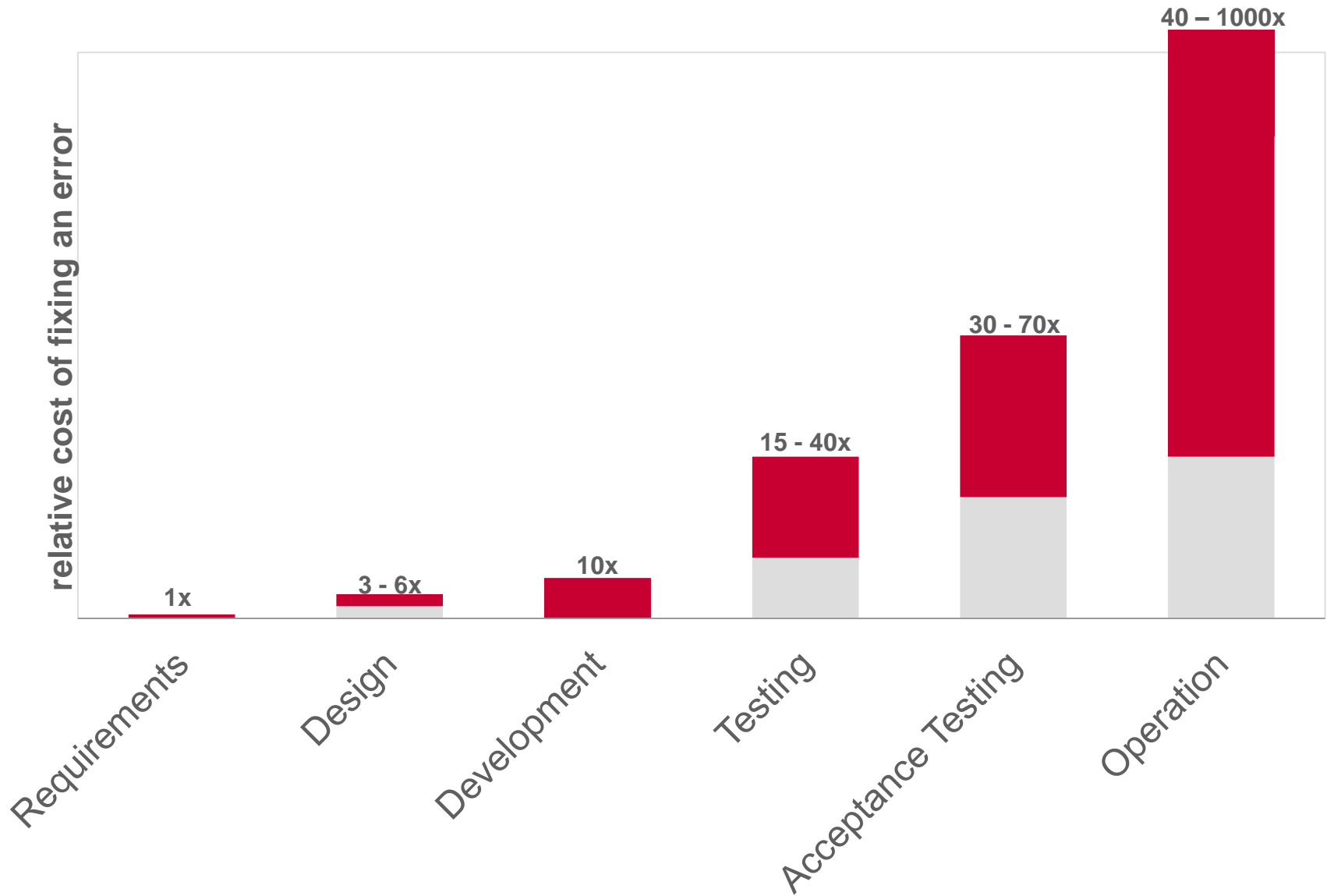
When will the world end
Tap to Edit >

**Well, Unix 32-bit time
overflows on January 19,
2038. Maybe then.**

Why is testing necessary?

- Testing was the **stepchild** of disciplines in software development
- Today, testing is a highly **mature, systematic, strategic** and **professional discipline** in software engineering
- **Driving the importance** of software testing today:
 - *Enormous growth in SW size and complexity*
 - *Enormous growth in SW integration level*
 - *Enormous growth in SW pervasiveness*
- Testing produces **high costs**, but SW failures are even more expensive

Potential Error Costs



Content

I Fundamentals

1 Importance of Testing

2 Definitions

3 ISTQB Testing principles

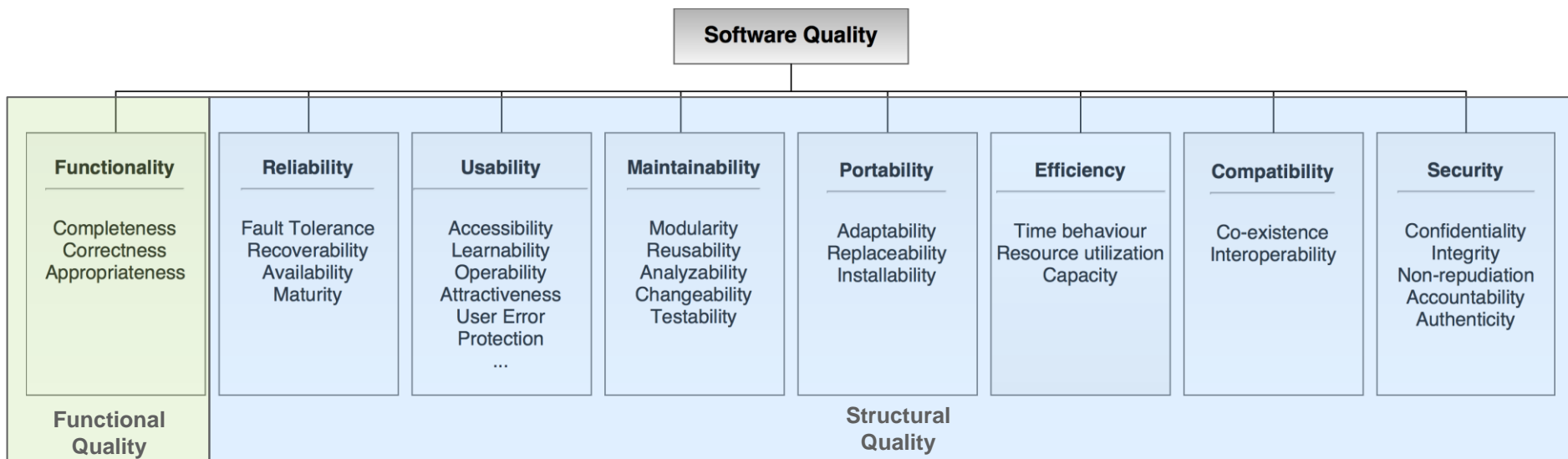
4 ISTQB Fundamental Test Process

5 Conclusion

Definition - Software Quality

- **Definition**

- The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs.
- Refers to the **product**, not the development **process**
- Set of characteristics (ISO/IEC 25010)



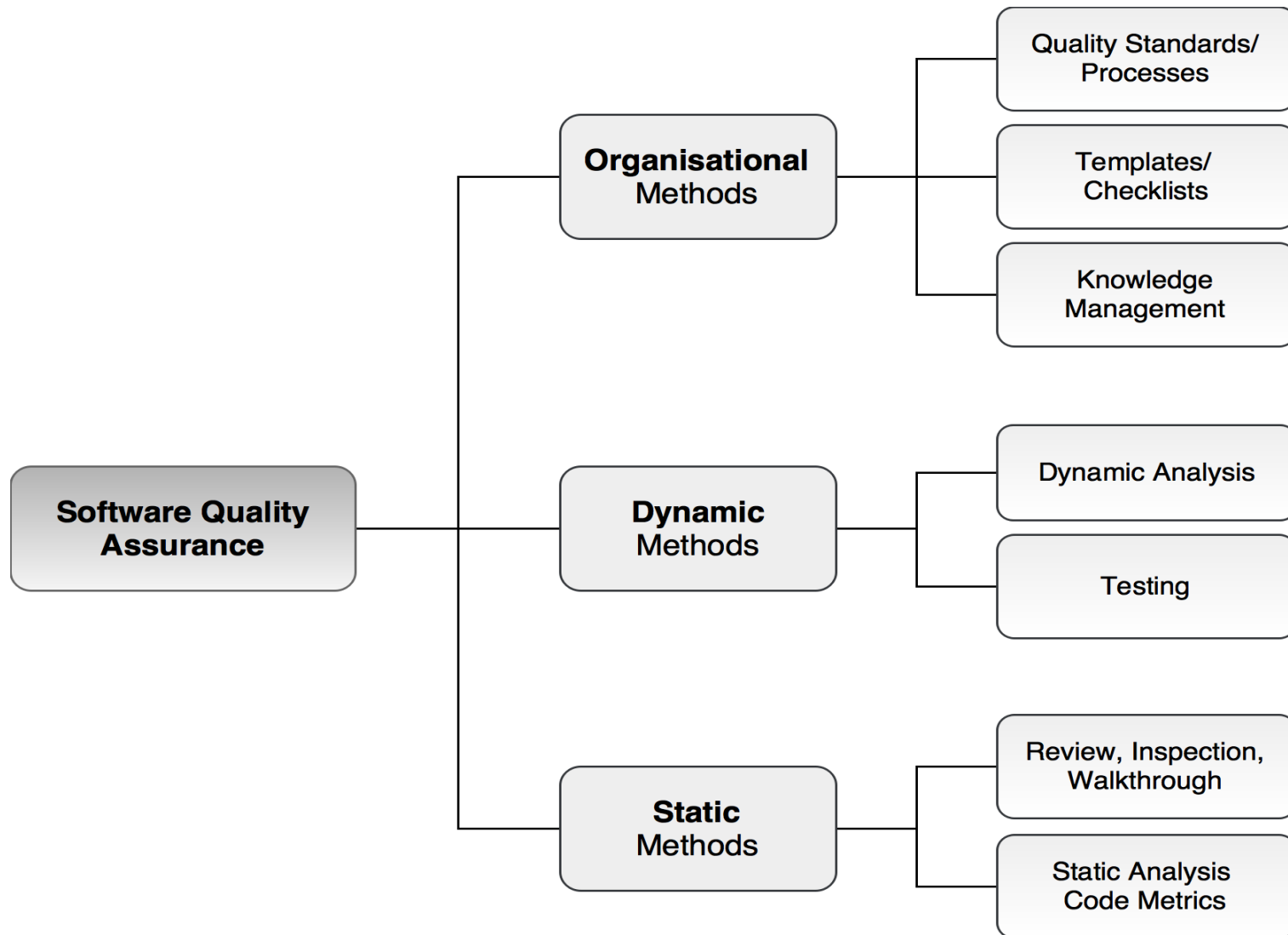
Definition - Software Quality Assurance

- **Definition**
 - Quality assurance is a **planned and systematic pattern** of all actions necessary to provide adequate confidence that an item or product conforms to **established technical requirements**. It is a **set of activities** designed to **evaluate** the process by which products are developed or manufactured. (IEEE 610.12-90)
- **On-going process** within the software development life cycle
- Supporting process to **verify requirements, keep time schedules** and **operate within the budget**
- **Testing** is an important **sub-area** of software quality assurance

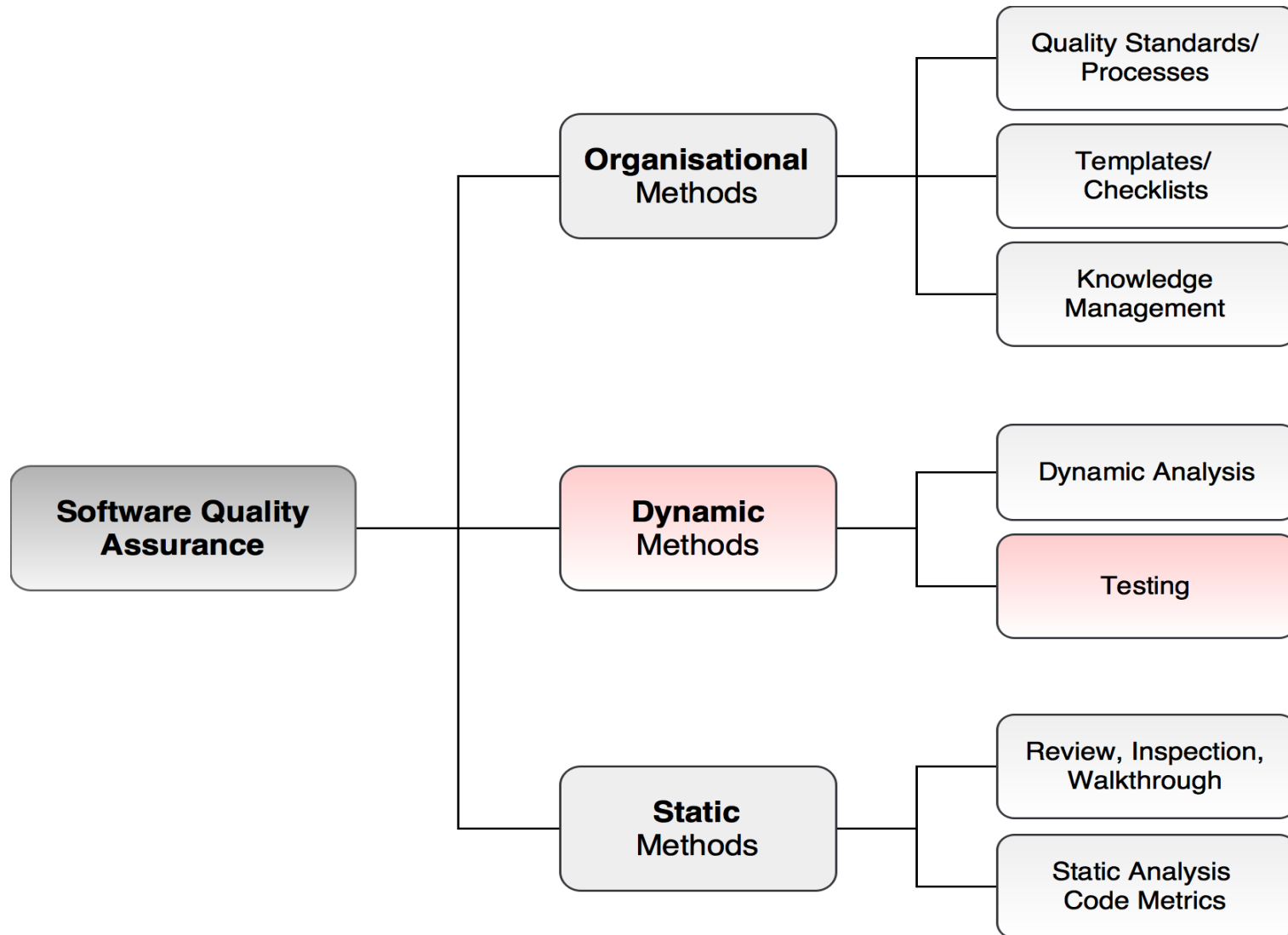
Definition - Software Quality Assurance

- **Organisational methods**
 - Provide an infrastructure (checklists, templates, guidelines, ...) to prevent or at least reduce the emergence of errors and quality deficiencies
- **Dynamic methods**
 - Evaluate the system at runtime (testing)
 - Monitor the system's behavior (debugging, trace analysis, log analysis)
- **Static methods**
 - Analyse a system without dynamic execution
 - Focus on metrics, control/data flow, code structure and design

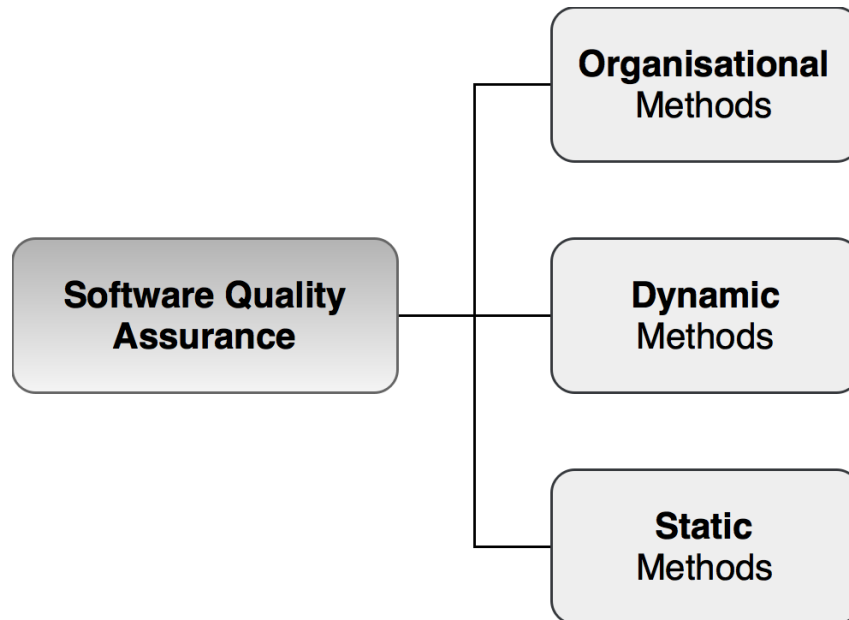
Software Quality Assurance



Software Quality Assurance



Software Quality Assurance



More detailed information on other software quality assurance methods in the summer-term course:

183.652 Software Quality Management

Definition – Software Testing

- **Definition**
 - Software testing is a **formal process** carried out by a specialized testing team in which a software unit, several integrated software units or an entire software package are examined by **running the programs** on a computer. All the associated tests are performed according to **approved test procedures** on **approved test cases**. (IEEE 610.12-90)
- Testing is an activity performed for evaluating the product quality, and for improving it, by identifying defects and problems.
- Software testing consists of the dynamic verification of the actual behavior of a program on a finite set of test cases against the expected behavior.

Definition – Errors, Faults, Failures

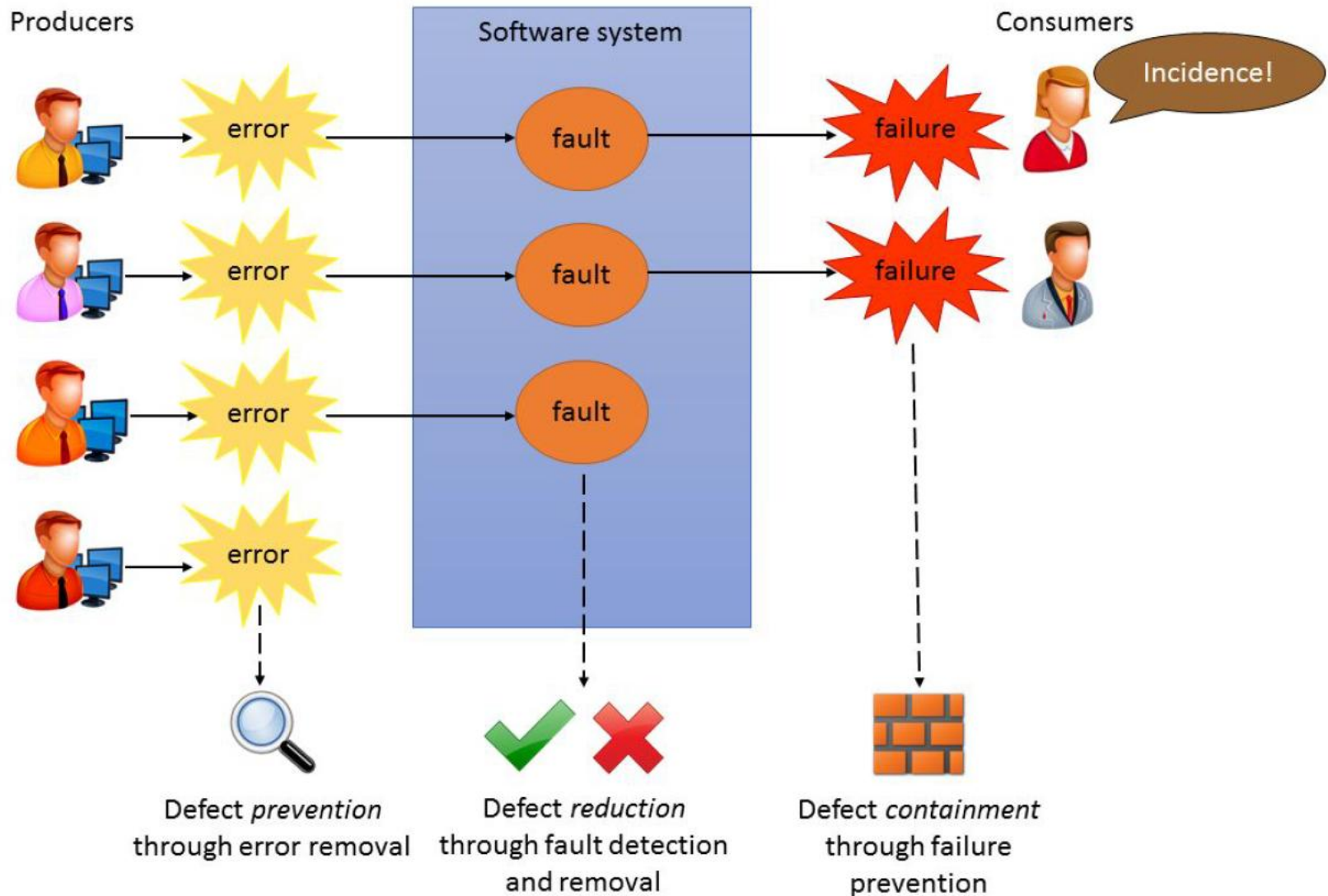
- **Software Errors**
 - Sections of code that are partially or totally incorrect as a result of a grammatical, logical or other mistake made by a systems analyst, a programmer, or another member of the SW development team.
- **Software Faults (Defects)**
 - Software errors that cause incorrect functioning of the software.
- **Software Failures**
 - Execution of faulty code triggered by a user's action.
 - The root of a failure is an error.

The cause of a failure cannot always be unequivocally identified.

Definition – Errors, Faults, Failures

- **Example**
 - Error
 - Developer forgets to implement a null check
 - Fault/Defect
 - Object is null at Runtime
 - Missing null check leads to NullPointerException at Runtime
 - Failure
 - Data is not stored after user clicks on save button

Definition – Errors, Faults, Failures



Testing vs. Debugging

- **Testing and debugging are distinct activities**
 - Testing is concerned with showing the existence of failures
 - Debugging is a development activity, concerned with locating, analyzing and removing the cause of failures
- The responsibility is usually
 - testers test
 - developers debug

Objectives

- Testing is the process of executing a program with the intent of
 - Finding defects
 - Preventing defects
 - Gaining confidence about the level of quality
 - Providing information for decision-making
- Testing is a constructive process
 - Does not intend to “destroy” the product
 - Finding a defect is no criticism
 - Communication needs to be constructive
 - Defects found during test will save time and money

Content

I Fundamentals

1 Importance of Testing

2 Definitions

3 **ISTQB Testing principles**

4 **ISTQB Fundamental Test Process**

5 **Conclusion**

- **International Software Testing Qualifications Board (ISTQB)**



- non-profit association
- software testing qualification certification organisation
- **ISTQB Certified Tester**
 - is a standardized qualification for software testers
 - different levels for progressively increasing learning objectives
 - Foundation
 - Advanced
 - Expert
- Lecture mainly covers foundation level contents

ISTQB testing principles

- **ISTQB testing principles**

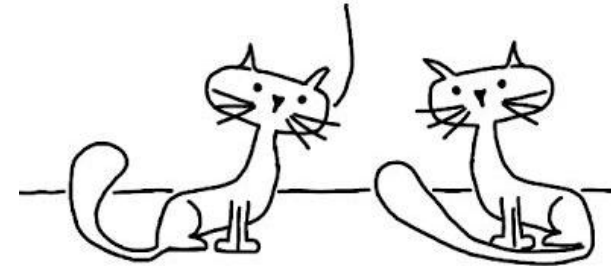
- #1 Testing shows presence of defects
 - but cannot prove that there are no defects

- #2 Exhaustive testing is impossible

- Testing everything is not feasible (all combinations of inputs and preconditions)
- Instead use risks-based decisions and priorities to focus testing efforts



I've checked every square foot in this house, I can confidently say there are no mice here.



- #3 Early testing

- Testing activities should start as early as possible in the software development life cycle and should be focused on defined objectives
- Testing early is a simple but effective strategy to reduce risks

Absence of proof is not proof of absence.
– William Cowper

- **ISTQB testing principles**

- #4 Defect clustering

- A small number of modules usually contains most of the defects.
- Testing effort should focus on the expected and observed defect density.

- #5 Pesticide paradox

- Tests that are repeated over and over again, will no longer find any new bugs

- #6 Testing is context dependent

- #7 Absence-of-errors fallacy

- Finding and fixing defects does not help if the system is unusable and does not fulfill the users' needs and expectations

Test activities

- Designing tests **early** in the life cycle can **prevent** defects from being introduced into the code
- Test activities exist **before and after test-execution**
- Test activities include
 - planning and control
 - choosing test conditions
 - designing and executing test cases
 - checking results
 - evaluating exit criteria
 - reporting on the testing process and system under test
 - finalizing or completing closure activities after a test phase

I Fundamentals

1 Importance of Testing

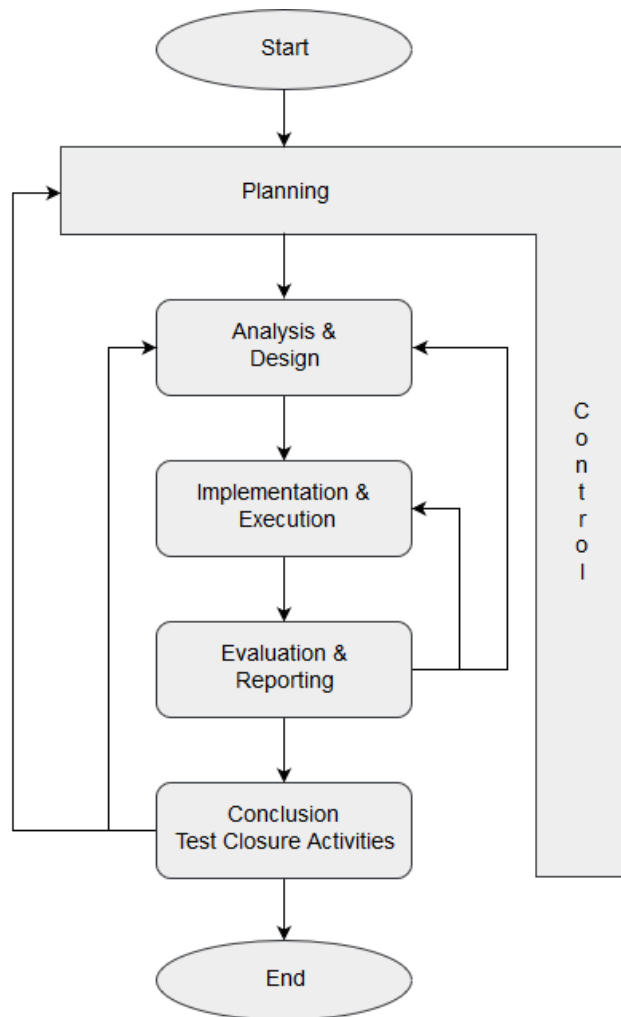
2 Definitions

3 ISTQB Testing principles

4 ISTQB Fundamental Test Process

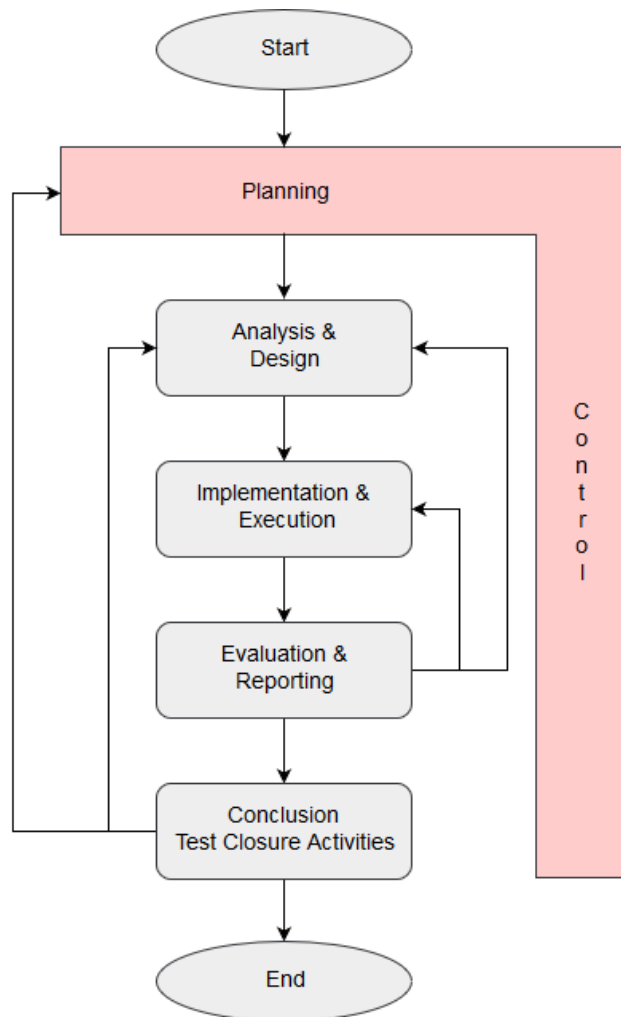
5 Conclusion

ISTQB Fundamental Test Process



- Testing is a process rather than a single activity
- Test execution is the most visible part of testing
- To be effective and efficient, testing also needs to be planned, prepared and evaluated
- Activities may overlap/take place concurrently
- The fundamental test process must be adopted to the project's needs

ISTQB Fundamental Test Process



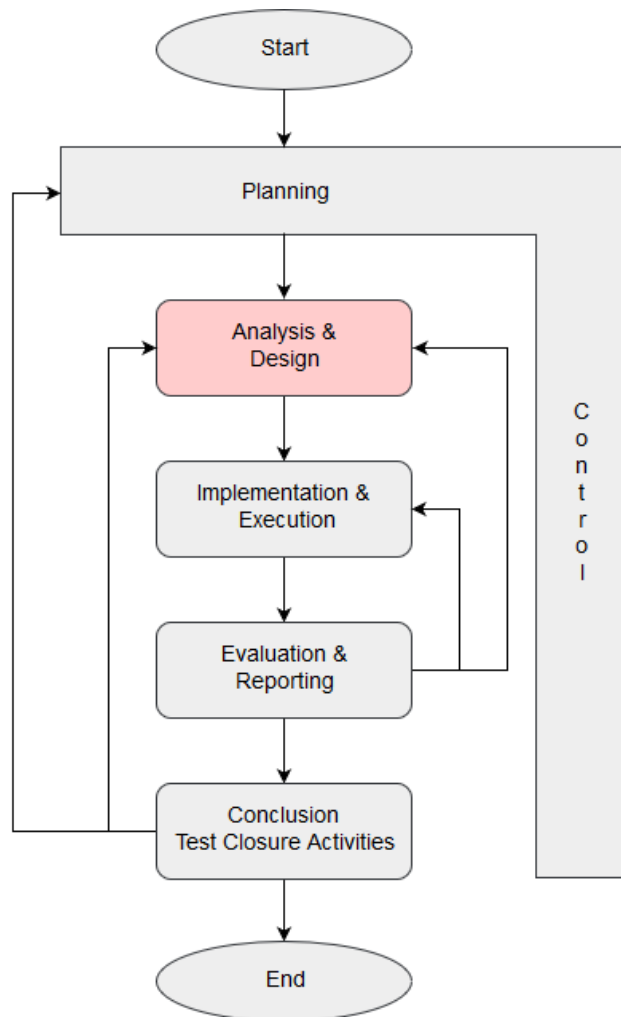
Test Planning

- Defining the objectives of testing and specifying the test activities
- Defining exit criteria

Test Control

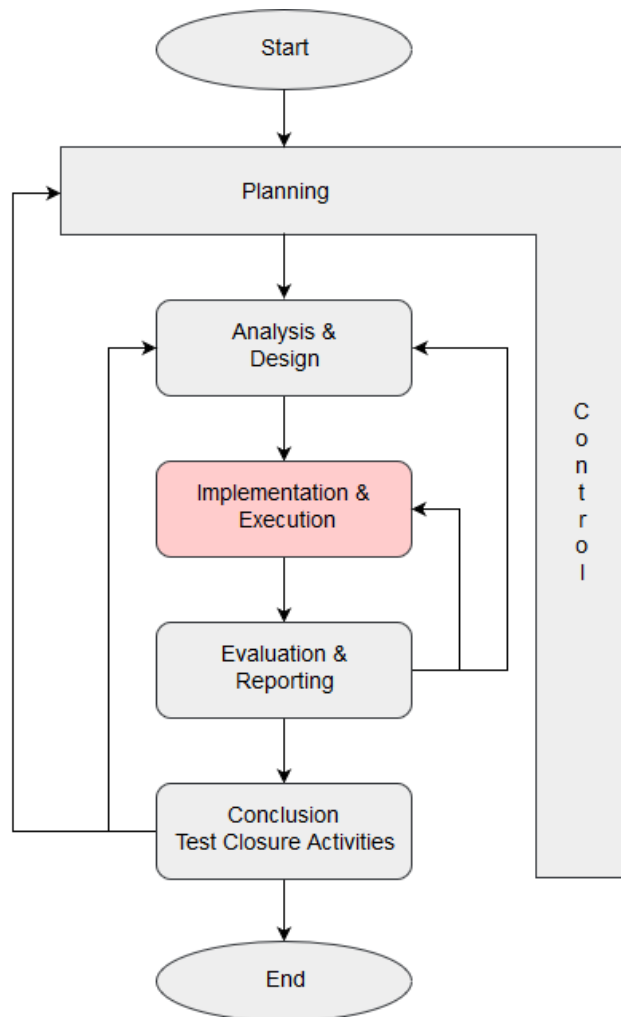
- On-going activity
- Comparing actual progress against the plan
- Reporting the status
- Taking necessary actions to meet the test objectives

ISTQB Fundamental Test Process



Analysis and Design

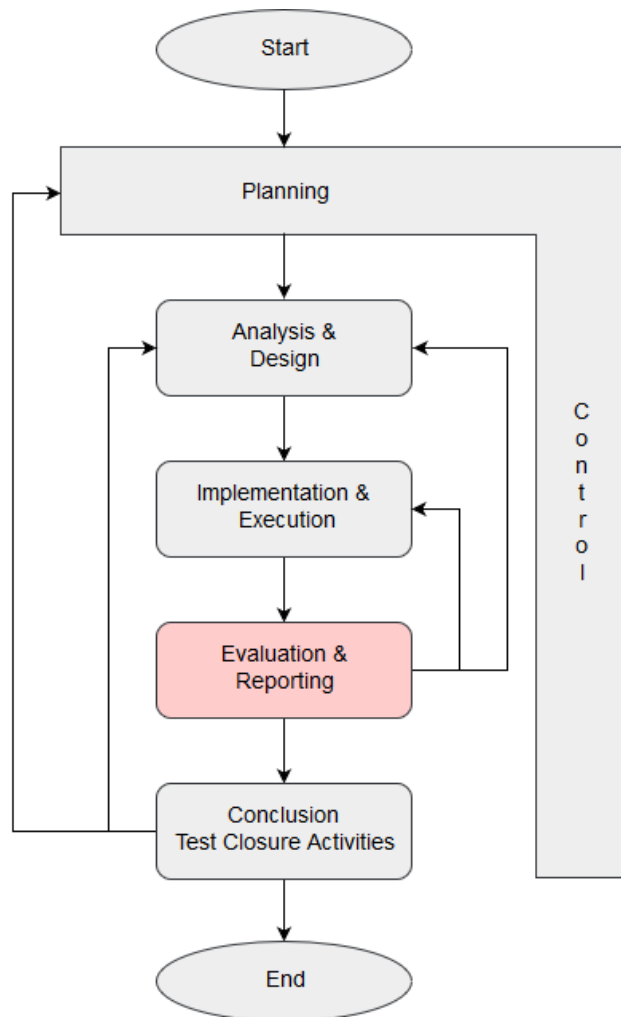
- Reviewing the test basis (requirements, design -, architecture-, risk-specifications)
- Designing and prioritizing high level test cases
- Identifying necessary test data to support the definition of test cases
- Creating bi-directional traceability between test basis and test cases
- Designing test environment and infrastructure and selecting tools



Implementation and Execution

- Developing and prioritizing test cases, creating test data, writing automated test scripts
- Creating test suites
- Comparing actual results with expected results
- Logging the outcome and reporting discrepancies as incidents
- Repeating test activities as a result of action taken for each discrepancy (re-test activities)

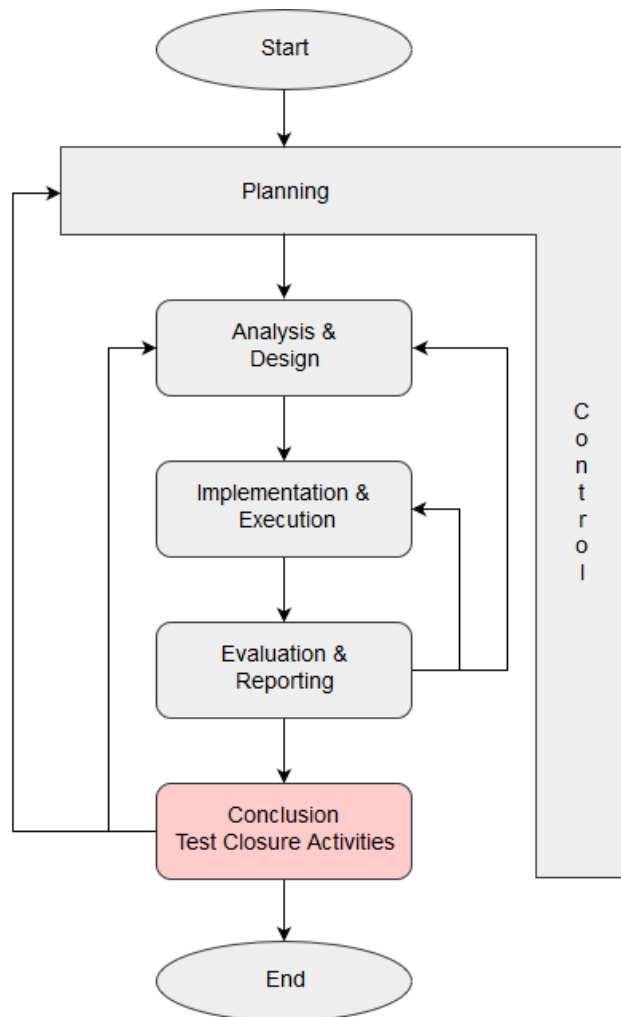
ISTQB Fundamental Test Process



Evaluation and Reporting

- Checking test logs against the exit criteria specified in test planning
- Assessing if more tests are needed or if the exit criteria should be changed
- Providing a summary report for stakeholders

ISTQB Fundamental Test Process



Closure Activities

- Occur at major project milestones (e.g. after a release, after a milestone is reached, after the end of a test project)
- Consolidate experience, facts and numbers
- Document lessons learned
- Handing over information to the maintenance team
- Using the gathered information to improve the test process

I Fundamentals

- 1 Importance of Testing
- 2 Definitions
- 3 ISTQB Testing principles
- 4 ISTQB Fundamental Test Process
- 5 Conclusion

Conclusion

- **Software Failures not only produce economical damage**
- **Testing produces high costs, but SW failures are even more expensive**
- **Testing is a process rather than a single activity**
- **Test activities exist before and after test-execution**
- **Testing activities should start as early as possible in the software development life cycle**

References

- T. Grechenig et al; Softwaretechnik Mit Fallbeispielen aus realen Entwicklungsprojekten, 2010
- ISTQB Foundation Level Syllabus
- A. Spillner and T. Linz; Basiswissen Softwaretest. 5. Aufl. dpunkt Verlag, 2012.
- D. Graham et al; Foundations of Software Testing: ISTQB Certification. Cengage Learning EMEA, 2008.
- IEEE Std 610.12-1990 (R2002), IEEE Standard Glossary of Software Engineering Terminology IEEE, 1990.
- A. Spillner et al; Praxiswissen Softwaretest – Testmanagement: Aus- und Weiterbildung zum Certified Tester, 2014
- B. Garcia; Mastering Software Testing with JUnit 5, 2017
- D. Galin; Software Quality: Concepts and Practice, 2018