OAMK

OULUN AMMATTIKORKEAKOULU

Software testing

Introduction to software testing

Pekka Ojala, Jouni Juntunen, 4.12.2025

Contents

- About software and testing
- Testing and quality assurance
- Static testing
- Dynamic testing
- Verification and validation
- AI and testing
- Role of testing
- Testing and costs
- Big bang testing
- Code and fix
- V model
- Agile testing
- Test driven development
- Role of tester in software project
- Sertifications

Software is everywhere

- Software is a part of everyday life, as it is used everywhere
- Software defects and malfunctions can lead to various consequences, such as…
  - Loss of money (for individuals and organizations)
  - Loss of time (for individuals and organizations)
  - Loss of reputation (for organizations)
  - Loss of life (for individuals)

Software testing in a nutshell

- Part of software development process, whose purpose is to find faults, deficiencies, and inconsistencies in the software
- A way to assess the quality of the software
- Testing can be manual, where a person tests the software by using it
- Testing can be automated, where tools—and increasingly artificial intelligence—are used to support the process
- Both, manual and automated testing, is needed
- Testing is performed against requirements, which means that they should be documented

Basic terminology

- Error/mistake
  - A human action that produces an incorrect result
  - E.g. a developer misunderstands a requirement and writes the wrong logic.
- Defect/bug/fault
  - A flaw in the software that may cause it to behave incorrectly.
  - E.g. a wrong conditional statement in the code.
- Failure
  - When the software behaves unexpectedly during execution.
  - E.g. The app crashes when the user presses a button.

Basic terminology

- Functional testing
  - o Testing the system’s **functions** against specified requirements
  - o E.g. Does the login form work?
- Non-functional testing
  - o Testing **attributes** like performance, security, usability, reliability
  - o E.g. Using a tool to measure performance of the developed software
- Regression testing
  - o Testing to ensure existing features still work after changes like bug fixes or feature update

Basic terminology

- Test environment
  - o The hardware, software, data, and configuration in which tests are executed
- Test execution
  - o Running the tests and recording results
- Test coverage
  - o A metric describing how much of the software is tested (e.g., code coverage, requirements coverage).
- Test level
  - o What “level” of the software is being tested (e.g., an individual function, the interaction of modules, or the entire system, …)

Basic terminology

- Test scenario
  - o High-level description of what needs to be tested
  - o E.g. Verify user can log in successfully.
- Test plan
  - o A document describing the testing goals, strategies, resources, schedule, and scope
- Test case
  - o A set of inputs, execution conditions, and expected results created to test a specific part of the software
- Test suite
  - o A collection of related test cases that are run together

Software testing and quality assurance

- Software testing has a close connection to quality assurance
- (Quality Assurance, QA)
    - o Process
    - o Extends beyond product and project boundaries
    - o Preventive action
- Software testing
    - o A set of tasks included in the software lifecycle
    - o Focuses on a specific product or project
    - o Both preventive and corrective action

Static testing

- Software is tested without actually using it
- Examples of different static testing methods
  - In a review, for example, the software's code or documentation is evaluated
  - In a walkthrough, the focus is on going through, for example, a software function that has been developed or is planned to be developed
  - In an inspection, software functions are checked based on a checklist
- The goal of static testing is to prevent defects from occurring

Dynamic testing

- Software is tested by using it
- The goal is to find faults, deficiencies, and inconsistencies in the softwar
- Testing can be manual or automated
- Testing is based on test cases, which tell the tester what and how to test, as well as the expected outcome
- Testing can also be exploratory, where the software is learned by using it
    - Test cases may not exist at first, but exploratory testing produces them so that testing can be focused on relevant aspects

Verification and validation

- **Verification and Validation (V & V)** are an essential part of software testing and quality assurance.
- **Verification** ensures that the product being developed meets the original design requirements and specifications.
  - *Are we developing the software correctly?*
- **Validation** ensures that the product being developed is suitable for its intended purpose and meets the needs and expectations of its future users.
  - *Are we developing the right software*

Software testing in the 2000s

- The importance of testing has continuously increased
- The more critical and extensive the system is, the larger the share of testing and bug fixing becomes
- The proportion of testing can rise to over 50% of all software development work done in a project or product development
- The maintenance phase of a software lifecycle may last a long time. Testing and bug fixing form a significant part of maintenance work.
- Software sometimes contains many defects and quality issues that can cause problems for end users

Software testing in the 2000s

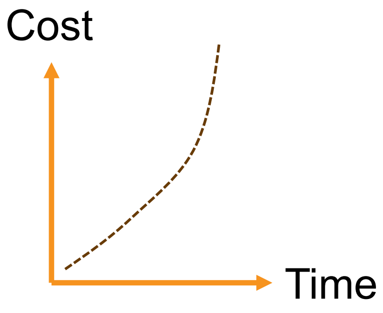
- Companies specializing in software testing have emerged in the software industry; these companies, for example, develop testing tools or offer testing services.
- Testing has become more diverse, as in addition to manual testing, testing is now performed automatically and based on various testing tools
- Manual testing is still needed because machines cannot replace humans
- The use of artificial intelligence is becoming more prominent in testing. AI is already being utilized, especially in automated testing

AI and testing

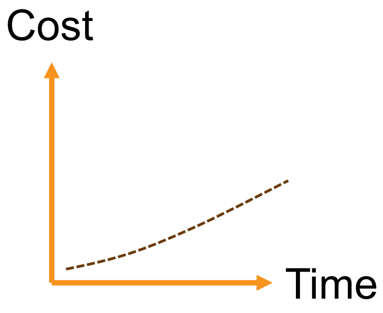
- Increasing usage of AI poses new challenges for testing
  - o Unpredictability and non-deterministic behaviour
  - o Data dependency
  - o Model can change ,there is no visibility how decisions are done (black-box)
  - o Requires lot of tests which must be updated constantly
  - o …
- Testing AI might require
  - o Dataset, model and adversarial testing
  - o Automation and monitoring
  - o …

Role of testing

- Testing plays an important role in ensuring software quality
- Testing shares common goals with other product development activities
- Testing should be seen as an integral part of the project or product development process from start to finish
  - One of the most common problems related to testing is that it is started too late
  - Testing is seen often as a separate support function that is included in product development too late and is not allocated enough time
  - The later software defects are discovered, the more expensive it is to fix them



If testing is started late or costs cut, in many cases fixing bugs will cause more costs later



If testing is started earlier and invested, it is more likely that cost of fixing bugs will stay lower

Big bang testing

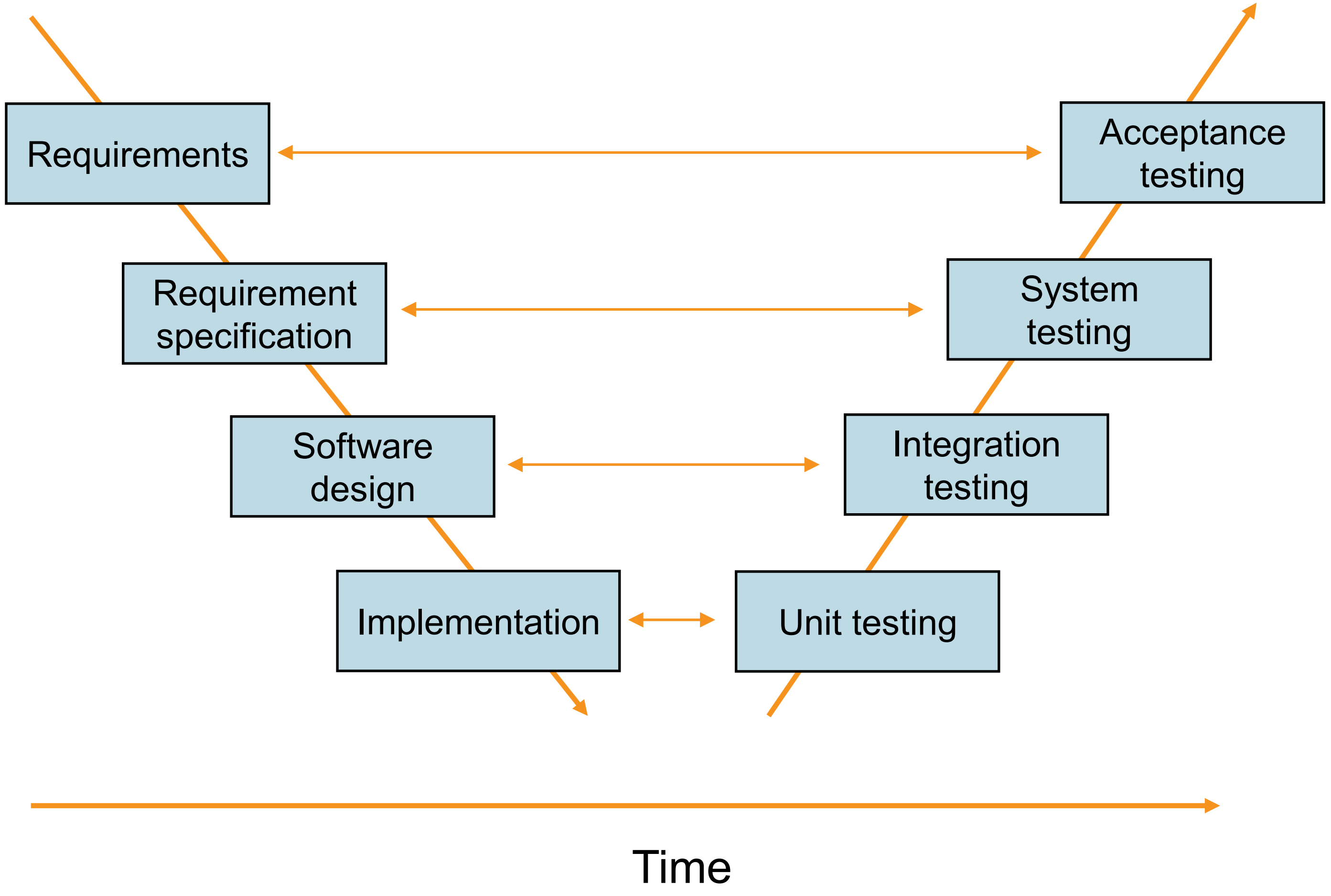
- Big Bang testing is when the entire software is tested at once
- Before release, only development-time testing is performed by programmer
- Different parts of the software are integrated into the Big Bang test
- There are no separate testing phases
- This approach is best suited for small, low-risk, non-complex projects
- It is not suitable for long-term development

Code and fix

- The "Code and Fix" model is where programmers fix defects as they discover them
- The person responsible for implementation is usually not the best person to do the testing
- This approach is suitable, for example, in the early stages of product development when a new product is being developed from scratch
- It is not suitable for long-term development, because at some point there will be a need for systematic testing

V model

- Used often if software process is following waterfall model
- Multiple phases of testing with its own specific purpose
- The prevalence of this approach has decreased as agile software development has become more common



Agile testing

- Agile software development addresses traditional testing problems better, such as starting testing too late
- Agile testing is usually used in conjunction with agile software development methods. Principles:
  - Testing is started early
  - Testing is done continuously
  - Testing is aimed to be automated
  - The goal of testing is to receive continuous feedback on software quality
  - The customer (or test users) actively participate in development and collaborate with developers

Test driven development

- Test-Driven Development (TDD) is a software development approach that belongs to agile methodologies. Principles:
    - o Programming starts with writing tests
    - o Tests are written based on the requirements, and at the same time, it is ensured that the requirements are understood
    - o Multiple tests are written, including tests that are expected to fail (this way, you can see how the application behaves in different situations)
    - o If there is no test written for a particular software function, it is assumed that the function does not work
    - o Every line of code is tested, and automation is relied upon, because it may be impossible to manually test every line of code in different situations

The role of tester

- The tester’s task is to try to find faults, deficiencies, weaknesses, and inconsistent behavior in the software
- The characteristics of the tester’s role become more pronounced if programmers and testers are different
    - The programmer develops the software by adding the required functionalities
    - The tester develops the software by trying to show that there are faults, deficiencies, weaknesses, or inconsistent behavior in the software

The role of tester

- Small software companies may not have full-time tester
- Large software companies or companies that provide testing services have full-time tester
- Good teamwork and communication skills are important qualities, as the tester works closely with various individuals and group

Designers

Product manager

Testers

Designers

Clients

The role of tester

- In agile software development, the tester is an expert in software quality issues and has, among others, the following tasks:
  - Helps programmers develop better software
  - Helps customers by explaining what is being sought through testing
  - Ensures good software quality
  - Provides feedback on software quality

Certification

- You can become certified as a tester, for example, through the FiSTB (Finnish Software Testing Board). More information: http://www.fistb.fi
- The most well-known certificate is the ISTQB (The International Software Testing Qualifications Board) Certified Tester Foundation Level (CTFL), which has two different versions
- All ISTQB certificates: https://www.istqb.org/#certifications-diagram

OAMK

OULUN AMMATTIKORKEAKOULU