# QA Fundamentals

## Introduction to Software Development


**SoftUni Team**

**Technical Trainers**

**Software University**

http://softuni.bg

# Table of Contents



# Have a Question?

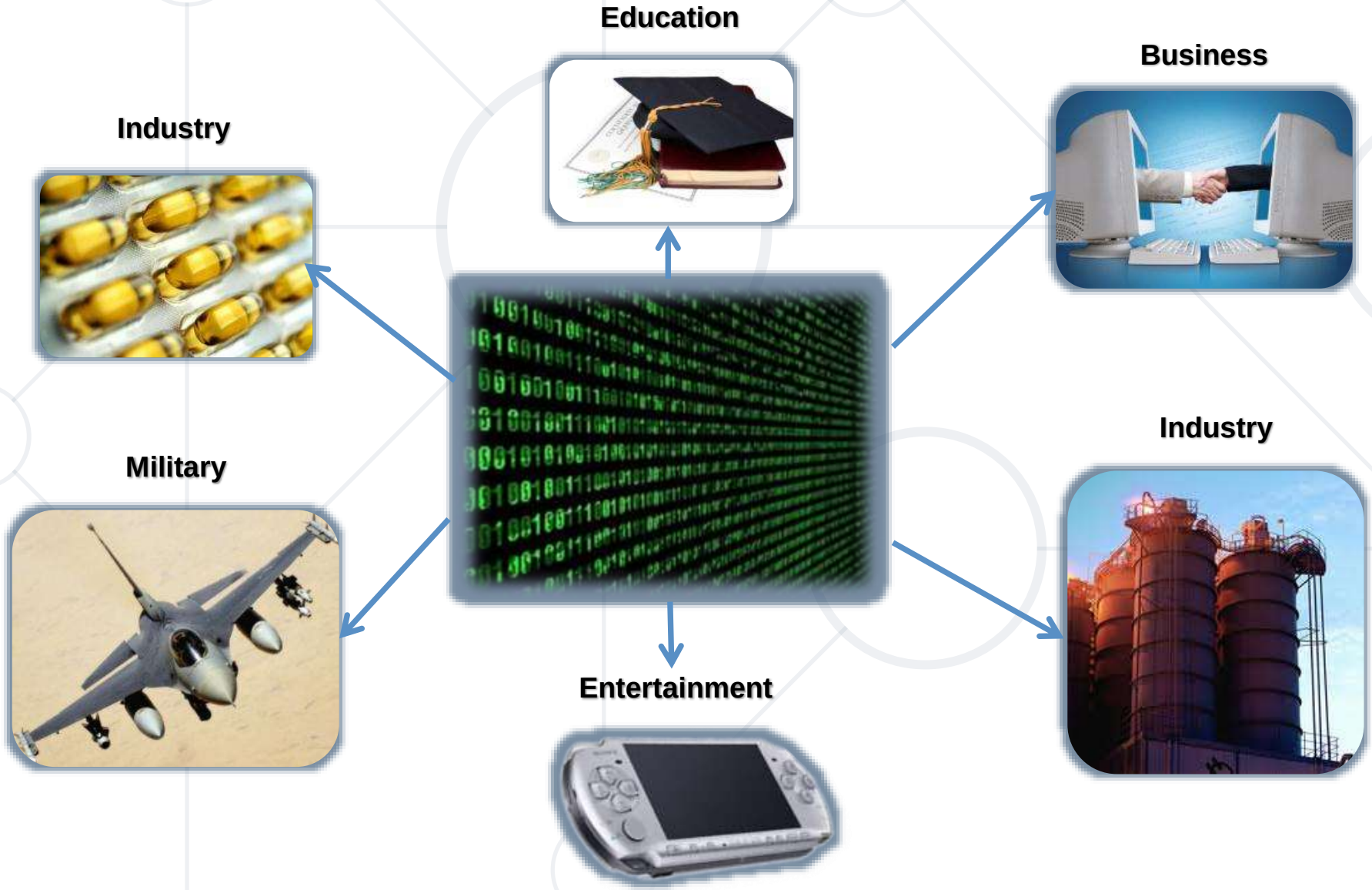
SoftUni Foundation

sli.do

#TECH-FUND

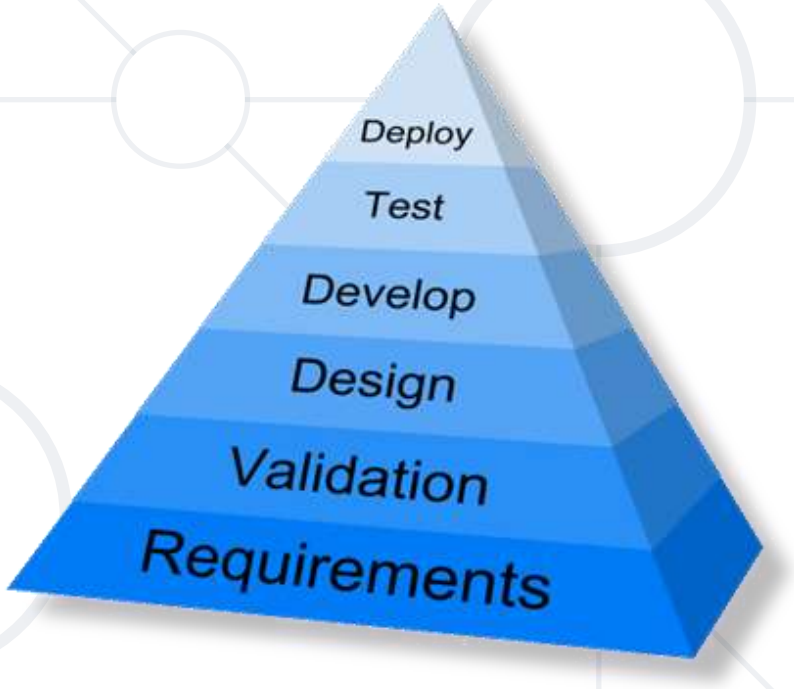
# Software Development

# First Look at Software



# First Look at Software (2)

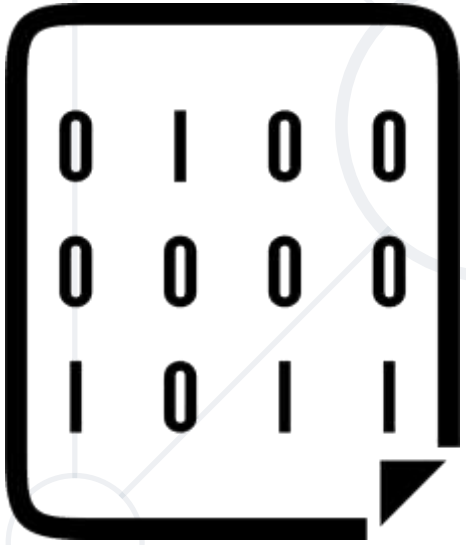
- Developed for 8 decades
- Evolved into a profession
- Concerned into quality maximizing
- Software engineering



# First Look at Software (1)

- Software Definition

> **Software is any set of machine-readable instructions that directs a computer's processor to perform specific operations**

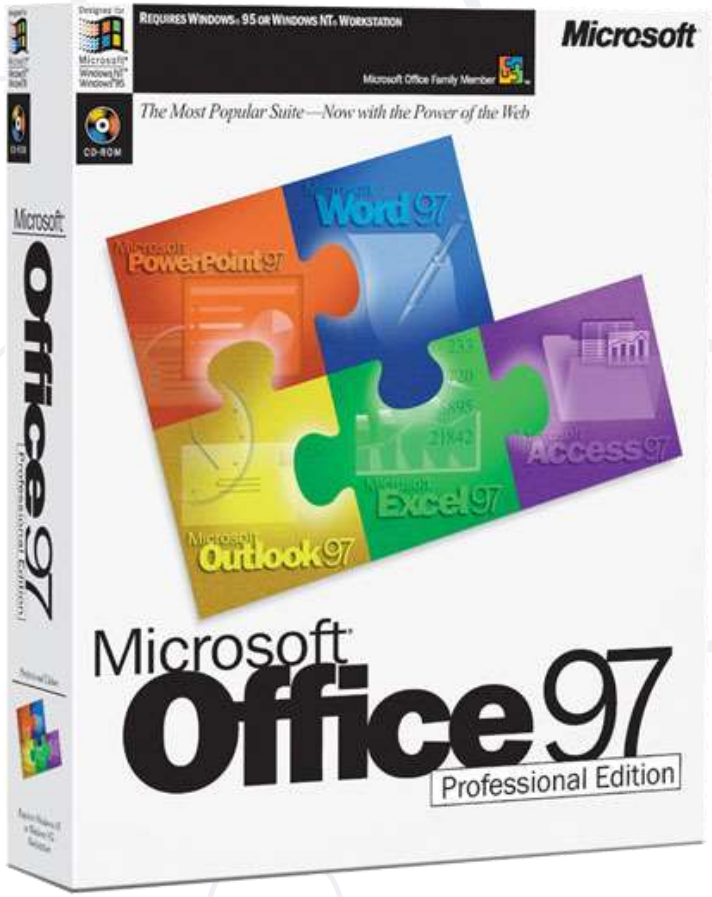


# Types of Software

## Software Distribution

# Retail Software

- The vast majority of software until the emergence of internet



# Web Applications


- The concept has been introduced in 1999 in Java Servlets
- A major leap in the software distribution
- Browsers and technologies are developed in favor of web apps
  - 2005 – the term Ajax was coined
  - 2011 – HTML5 was finalized


# Mobile Applications

- App distribution platforms began appearing in 2008
  - Apple app store
  - Google play
  - Windows phone store



- Started as productivity and information retrieval apps
- Reached bigger market share than retail software

# Bugs!

# Software Defects

- Human being can make an **error (mistake)**
- Errors produce **defects**
  - Defects are **faults / bugs** in the program code, or in a document
- If a defect in code is executed, that might cause a **failure**:
  - Fail to do what it should do
  - Do something it shouldn't

# Software Defects

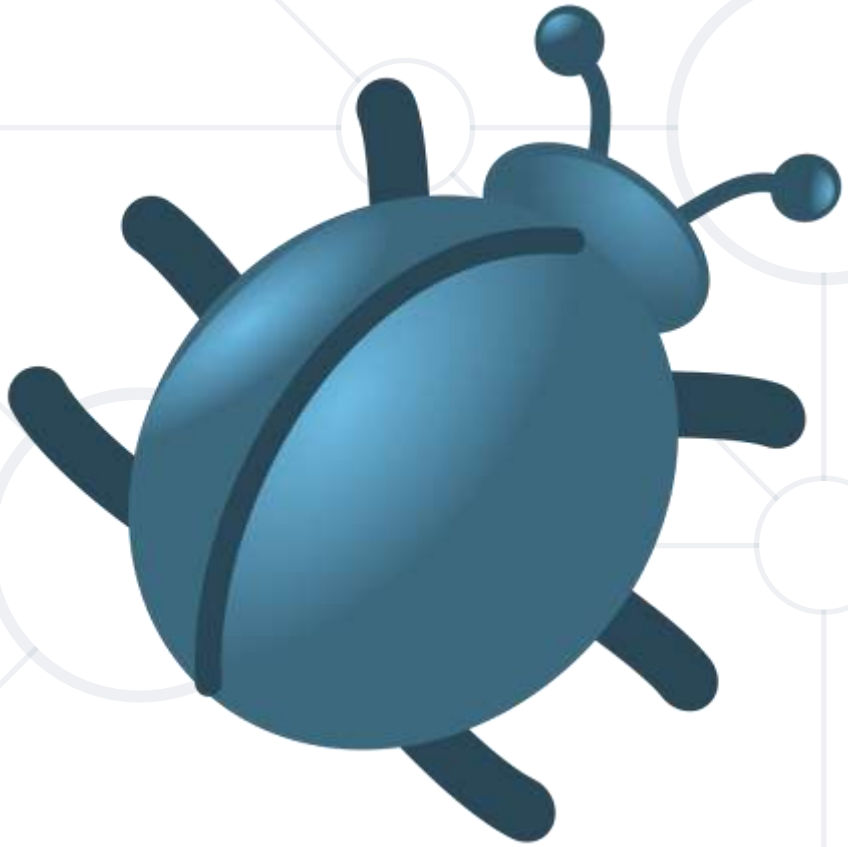
- The human factor
  - Humans make **mistakes**
  - Poor **training**
  - **Time** pressure
  - **Code** complexity
  - Complexity of **infrastructure**
  - Changing **technologies**

# Software Defects

- Organizational factors
  - Inefficient **communication**
  - Unclearly defined **requirements**
- Environmental conditions
  - Radiation, Magnetism, Electronic fields, Pollution, Etc.
  - These can change the hardware conditions

# Software Defects

- Anomaly
- Error
- Bug
  - Defect
  - Fault
  - Problem
- Failure
- Defect/fault masking



# Bug Fixing Importance

# Bug Fixing importance (2)

# Bug Fixing Importance (3)

- Unfixed bugs camouflage other bugs
- Suggest quality isn't important
- Duplicate effort
- Unreliable metrics and money loss
- Fixing a bug today costs less than tomorrow

# Bug Fixing Importance

## What Is Testing?

# What Is Testing?

- The process of **exercising** software
  - To verify that it satisfies specified requirements and to detect errors
- The process of **analyzing** a software item
  - To detect the differences **between existing and required conditions** (that is, bugs)
  - To **evaluate the features** of the software item

# What Is Testing?

- The process of **operating** a system or component under **specified conditions**
  - Observing or recording the **results**
  - Making an **evaluation** of some aspect of the system or component

# Main Test Activities

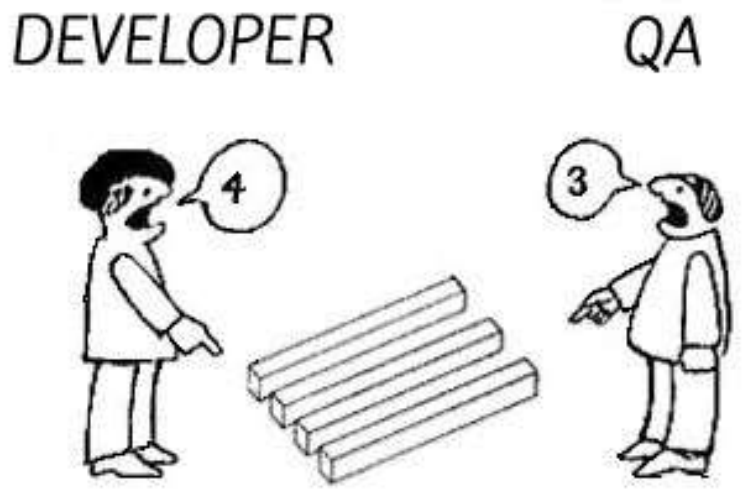
- Testing is **not just running tests**, but also:
  - Planning and control
  - Choosing test conditions
  - Designing and executing test cases
  - Checking results
  - Evaluating exit criteria
  - Reporting on the testing process and system under test
  - Finalizing or completing closure activities

# Main Test Objectives

- Testing pursues several **objectives**:
  - Finding defects
  - Gaining confidence about the level of quality
  - Providing information for decision-making
  - Preventing defects

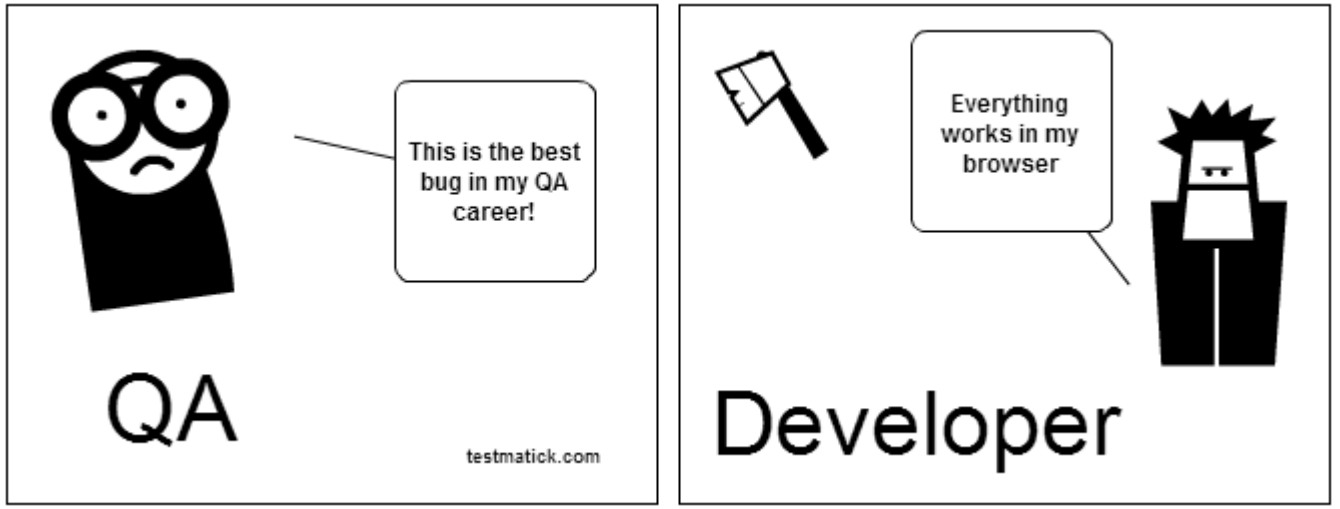
# QA vs Devs

- QA testers
  - Are **perceived as destructive** – only happy when they are finding faults!
  - Usually require **good communicatio n skills, tact & diplomacy.**
  - Normally need to be **multi-talented** (technical, testing, team skills).



# QA vs Devs

- Developers
  - Are **perceived as** very **creative** – they write code without which there would be no system
  - Are **rarely good communicators**
  - Can often **specialize** in just **one or two skills** (e.g. VB ,C++,JAVA,SQL)

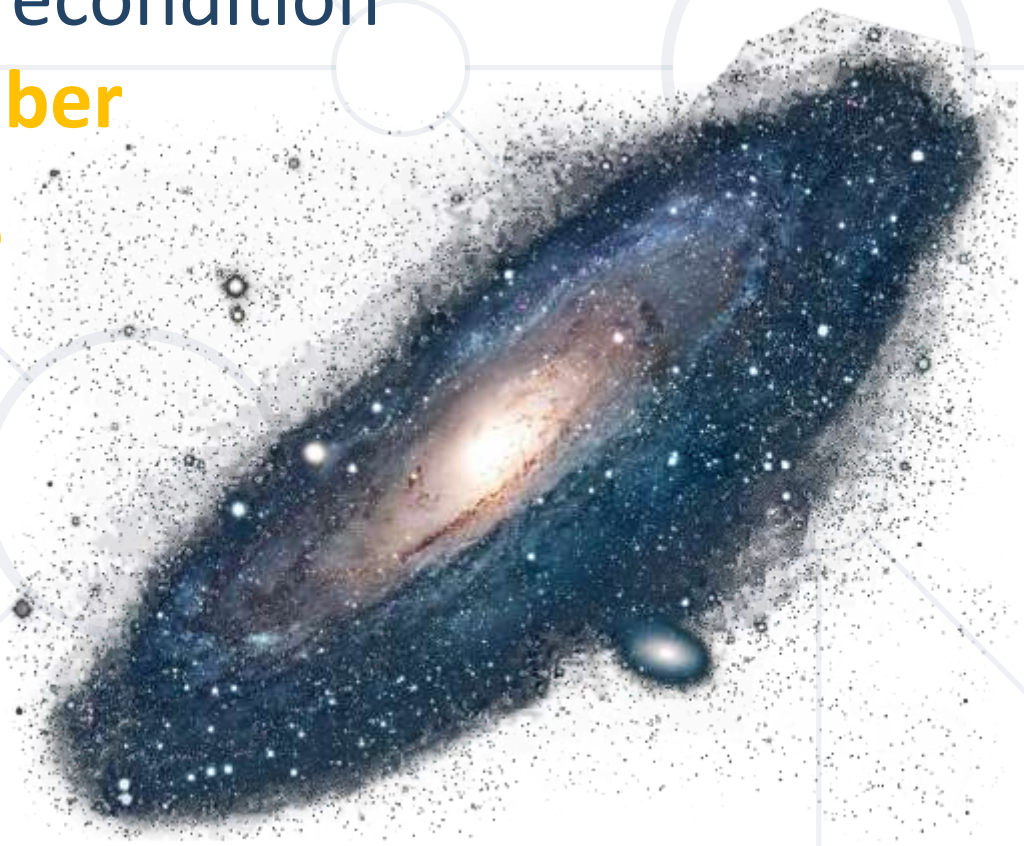


# Seven Testing Principles

- Testing shows presence of defects
  - Testing can show that **defects are present**
  - Cannot **prove** that there are **no defects**
  - Appropriate testing **reduces** the probability for **defects**

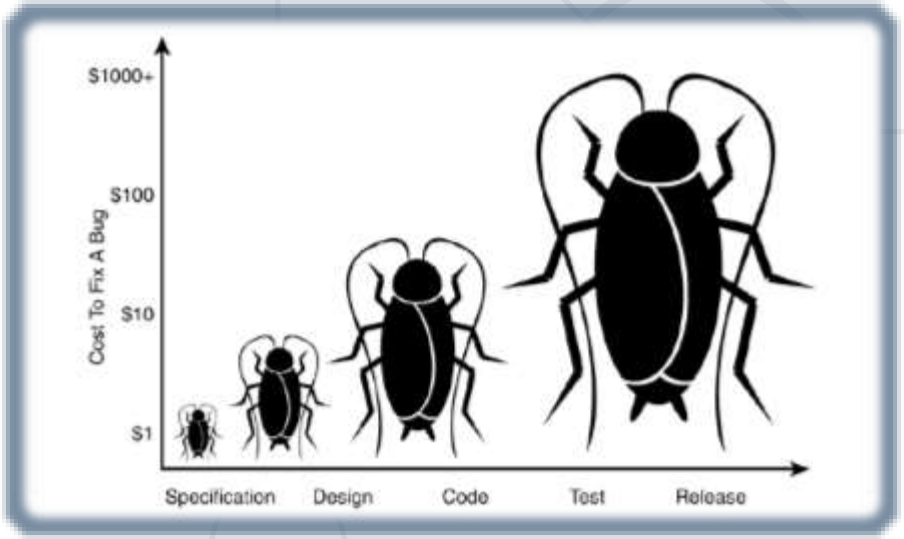
# Seven Testing Principles (2)

- **Exhaustive testing is impossible**
  - All combinations of inputs and precondition s are usually **almost infinite number**
  - Testing everything is **not feasible**
  - **Risk analysis and priorities** should be used to focus testing efforts



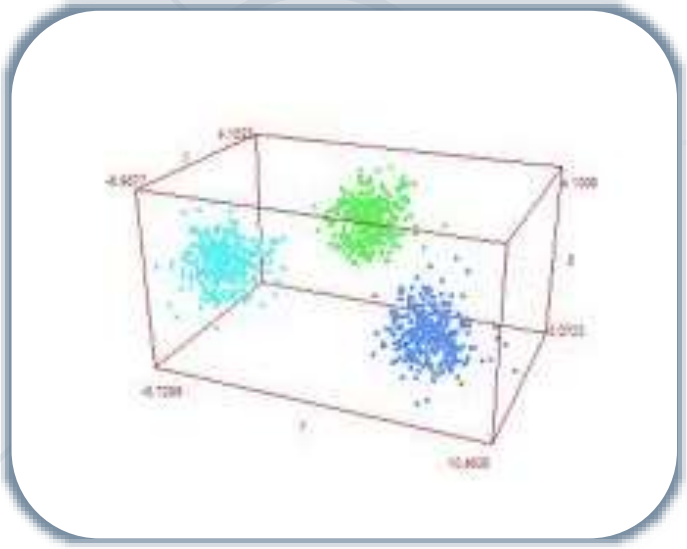
# Seven Testing Principles (3)

- **Early testing**
  - Testing activities shall be started **as early as possible**
    - And shall be focused on defined objectives
  - The later a bug is found – **the more it costs!**



# Seven Testing Principles (4)

- **Defect clustering**
  - Testing effort shall be **focused proportionally**
    - To the expected and later observed defect density of modules
  - **A small number of modules** usually contains most of the defects discovered

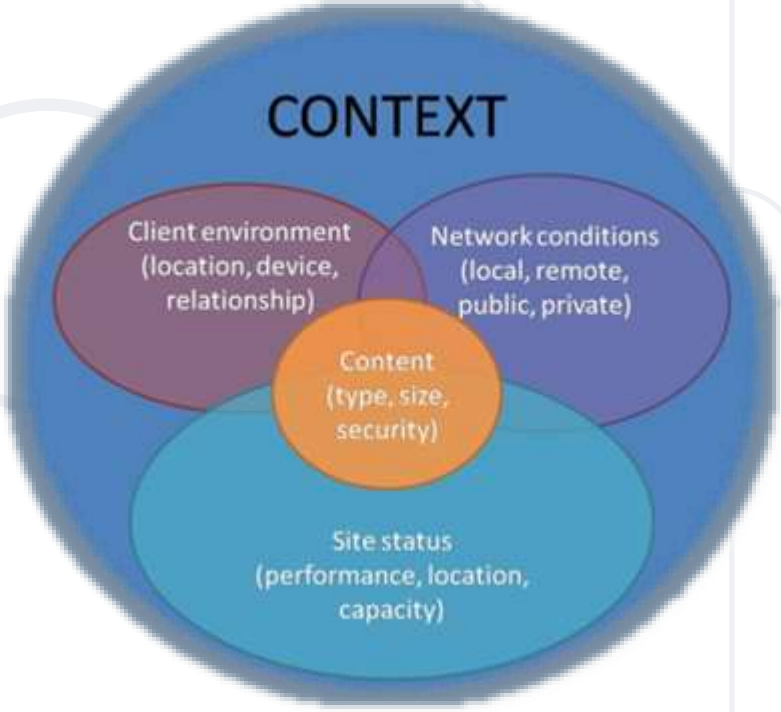


# Seven Testing Principles (5)

- **Pesticide paradox**
  - Same tests **repeated** over and over again tend to **loose their effectiveness**
    - Previously undetected defects remain undiscovered
  - New and modified test cases should be developed

# Seven Testing Principles (6)

- Testing is context dependent
  - Testing is done differently in different contexts
  - Safety-critical software is tested differently from an e-commerce site



# Seven Testing Principles (7)

- **Absence-of-errors fallacy**
  - Finding and fixing defects itself does not help in these cases:
  - The system built is **unusable**
  - Does not fulfill the **users needs and expectations**

# Summary

- Software evolved through the time
  - Retail, web, mobile
- It still has bugs
- Reasons does not change
- Importance of fixing bugs
  - More Customers
  - More money
  - No unsatisfied developers

# Literature

- QA Book: www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html4
- Tutorials: www.w3schools.com
- Selenium: leanpub.com/selenium-webdriver-book
- GitHub: github.com/QualityAssuranceTeam/source
- More Tutorials: www.codecademy.com

# Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

# SoftUni Diamond Partners

Software University













# SoftUni Organizational Partners

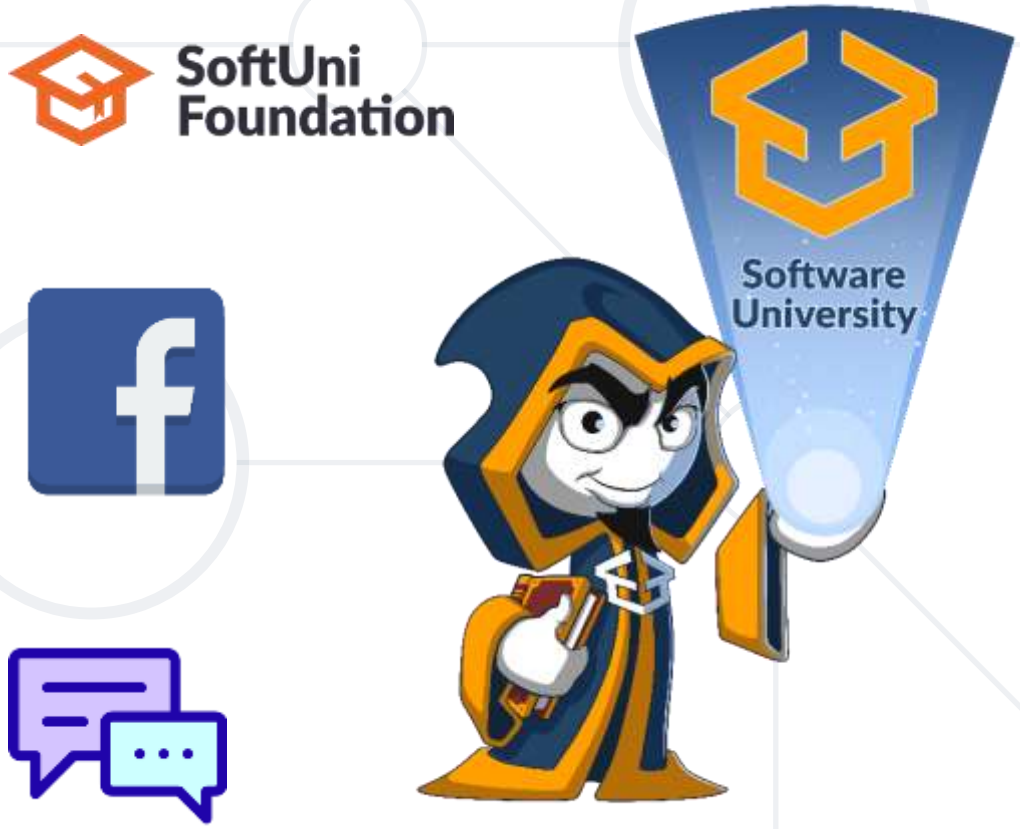
Software University





# Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg



# License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license