**Testing Introduction**



**Software University**

[**http://softuni.b**](http://softuni.bg/)

[**g**](http://softuni.bg/)

**SoftUni**

**Team**

**Technical Trainers**



Seven Testing Principles and Test Process



1.

What is Testing?

2.

Importance of Software Testing

3.

Seven Testing Principles

4.

~~t~~

Process

Tes

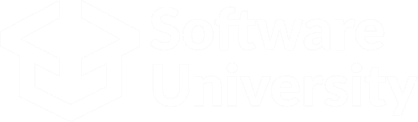
5.

Psychology of Testing

6.

Software Development and Testing

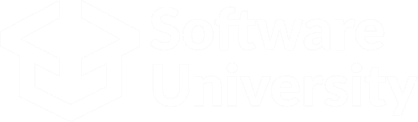
2



**Table of Contents**

# sli.do

**#QA-Fund**



**You Have Questions?**

3

**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



To evaluate the features of the software item

**What is Testing?**



6





Software Testing is a way to



Assess the

**quality**

of the

**software**



**Reduce**

the

**risk**

of software failure in operation



It includes



Test

**planning**

and

**analyzing**



**Designing**

and

**execution**



**Reporting**

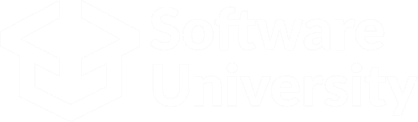
and

**evaluation**

**Software Testing**



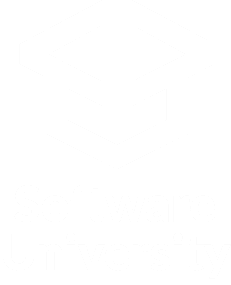
* Main **objectives** of testing
* **Prevention** of **defects**
* Verification of all **specified requirements**
* Verification of the **expected behavior** of a software



**Testing Objectives**

* To **reduce** the **level of risk** of inadequate software
* To provide **information** for stakeholders
* To comply with **contractual**, **legal** or **regulatory** requirements

caused by defects in





**Debugging**

is a

development activity that

finds and fixes defects



Debugging is usually done

by

**developers**



**Testing**

shows failures

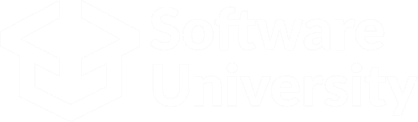
software



**Testers**

are usually

responsible for testing



**Testing vs. Debugging**

## What Could Cause Incidents? (1)

* Reasons for the existence of defects / bugs



The cause of human mistakes might be



**Time**

pressure



Code

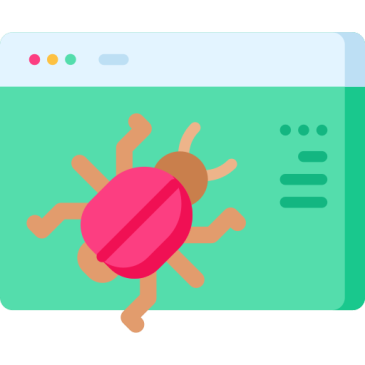
**complexity**



* **Human beings** can make **errors** (mistakes)
* Poor **training**
* Complexity of infrastructure
* Changing **technologies**

## What Could Cause Incidents? (2)

* **Organizational** factors
* Inefficient **communication**
* **Unclearly** defined requirements
* **Environmental conditions**
* Radiation, magnetism, electronic fields, pollution, etc.
* These can change the hardware conditions



## What Could Cause Incidents? (3)

* Other causes of incidents include:

Misconfiguration or failure of the **test environment**



**Corrupted**

test data



**Bad**

tests



**Invalid**

expected results



**Tester**

mistakes

* According to the test policy any type of incident can be logged for tracking

10

**Importance of Software Testing**





Software Testing contributes to the

**quality**

of the

**components**

or

**systems**



Makes sure that the software meets contractual or

legal requirements



Software testing could

**reduce costs**

significantly



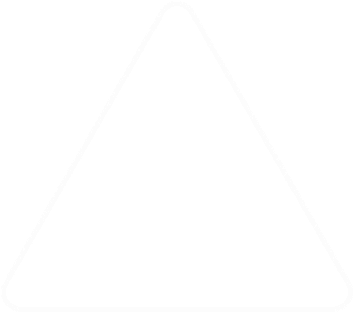
Testing of software can

**save**

**~~l~~**

**ives**

**Importance of Software Testing**



12

**Most Crucial Software Bugs (5)**

Mars Climat Orbiter Crash (1998)



Designed to study the Martian climate,

atmosphere, surface



Expected to be on 140

-

150

km from the

planet



Reached 57km and was destroyed from

the pressure



COST: $125 million

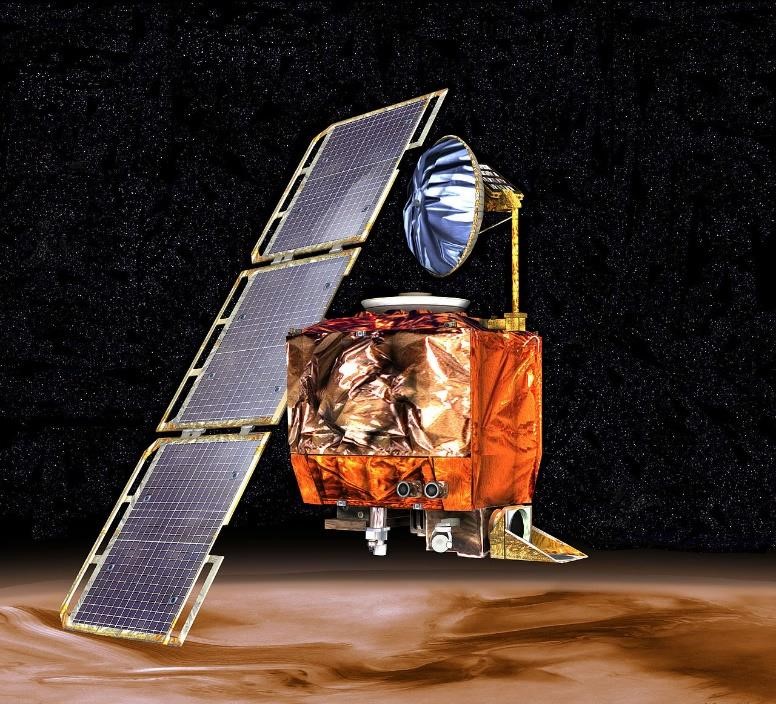
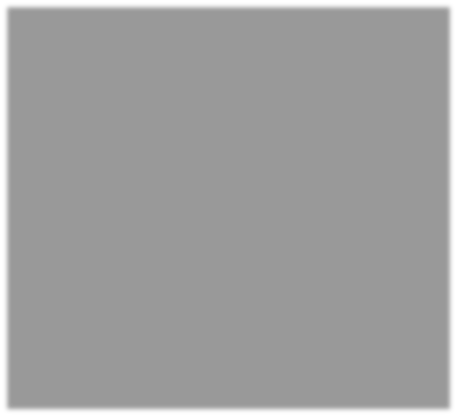


Reason: Incorrect measurement usage

(

imperial versus the NASA specified ones

)



14

**Most Crucial Software Bugs (4)**

Mariner I space probe (1962)



Mariner 1 was the first spacecraft of the

Americans Mariner program



Designed to explore Venus



Faulty application of the guidance commands



Destroyed 5 minutes after launch



COST: $18.2 million



Reason: The overbar's resemblance

to a hyphen

(

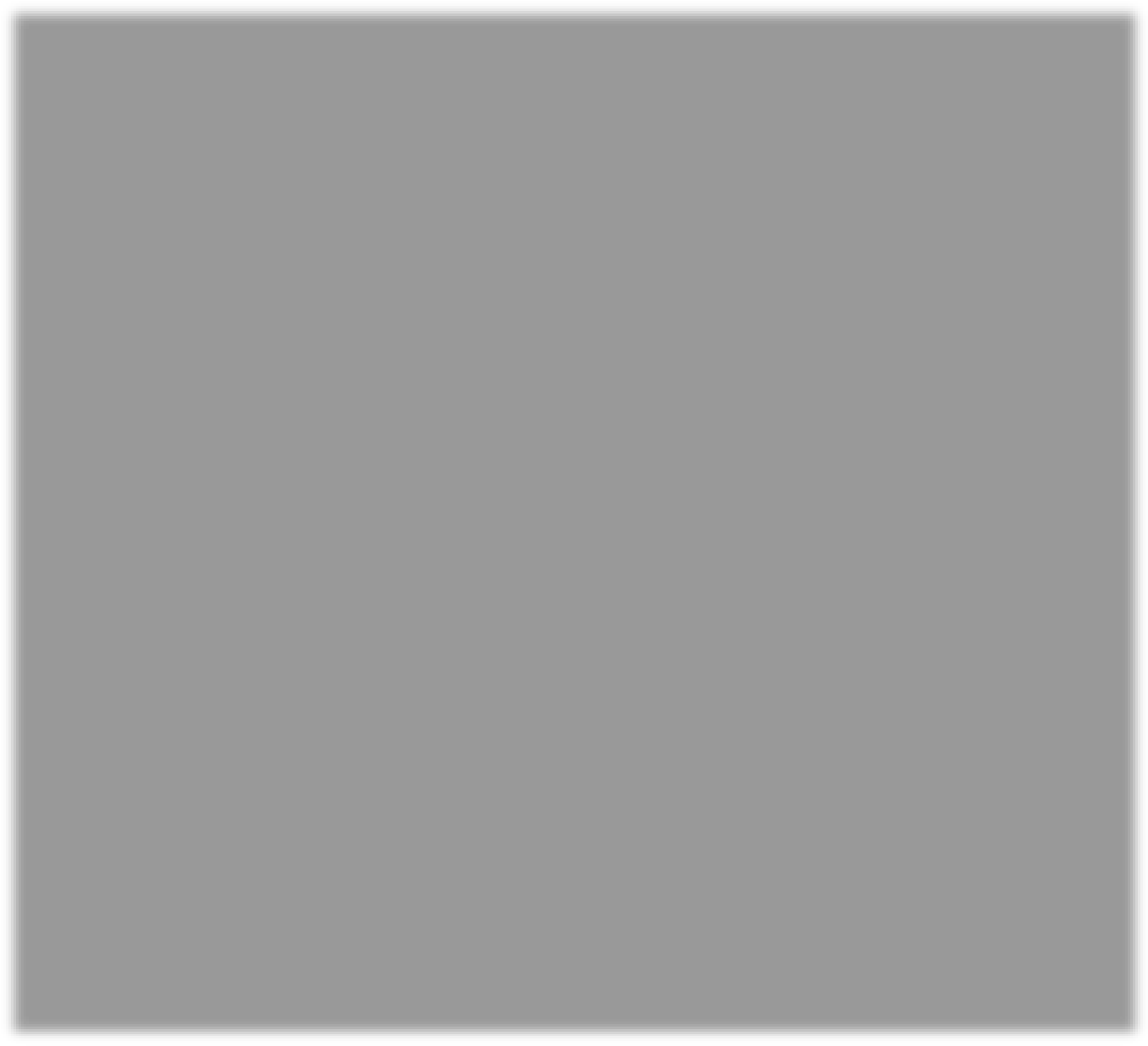
'‾'

versus

'

-

')



**Most Crucial Software Bugs (3)**

Ariane 5 Flight 501 (1996)



Europe’s newest unmanned satellite

-

launching

rocket



Destroyed 36.7 seconds after launch



COST: $8 billion with carrying

$500 million satellite



Reason: Softwa

~~r~~

e tried to cram 64

-

bit

number in 16

-

bit space which resulted

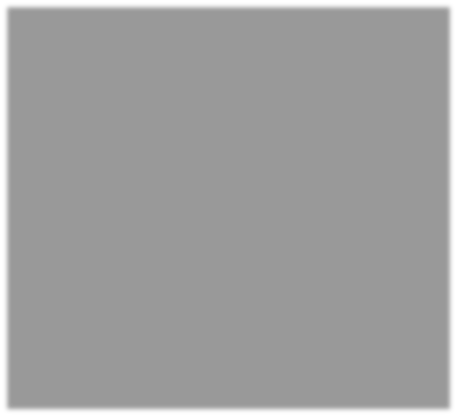
in overflow conditions



Video:

[https://youtu.be/qnHn8W1Em6](https://www.youtube.com/watch?v=qnHn8W1Em6E)

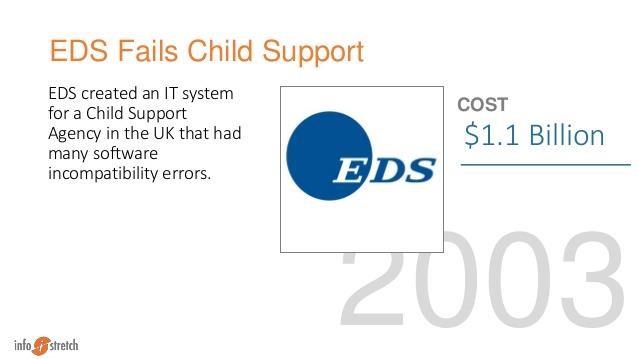
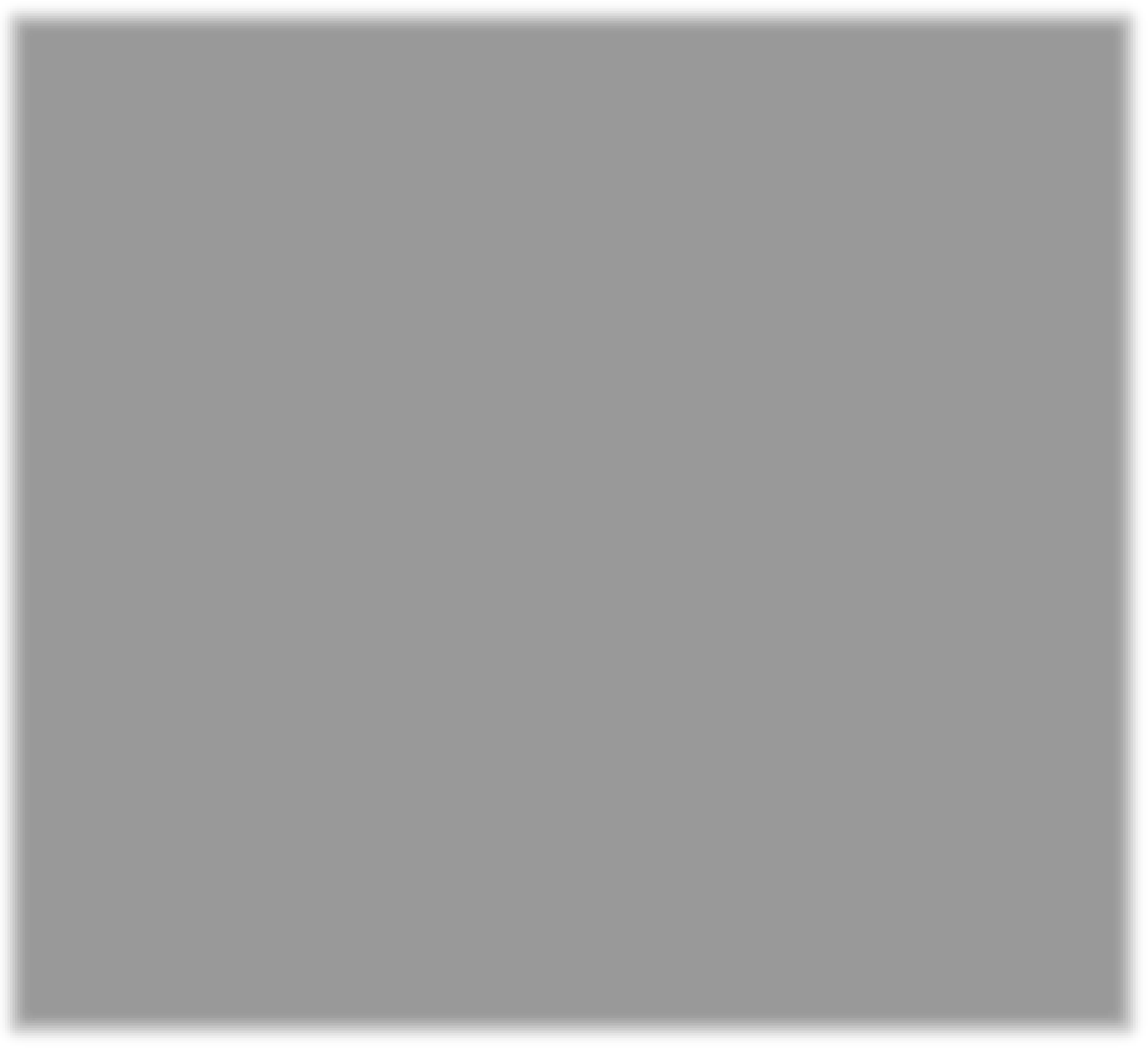
[E](https://www.youtube.com/watch?v=qnHn8W1Em6E)



### Most Crucial Software Bugs (2)

EDS Fails Child Support (2004)

* IT system for the UK Child Supporting Agency
* Department for Work and Pensions restructured the entire agency
* Incompatibility between the restructure and the new software - 500 bugs reported

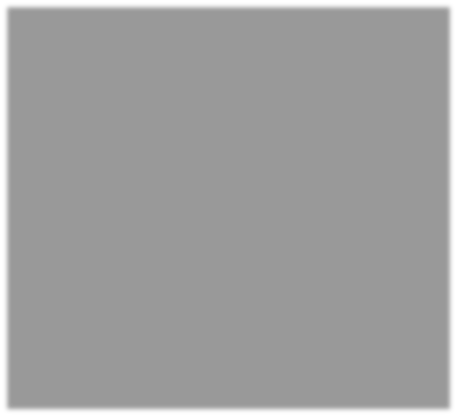


* Result: Overpay 1.9 million people, underpay another 700K, had $7 billion in uncollected child support payments
* COST: $1.1 billion

### Most Crucial Software Bugs (1)

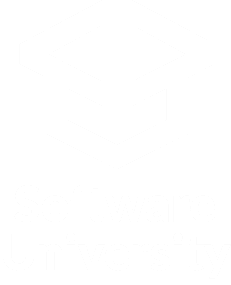
Soviet Gas Pipeline Explosion (1982)

* CIA discovered that Soviet Union was trying to steal sensitive US technology for its operations of their trans-Siberian pipeline



* CIA introduced bug into the Canadian built system (expected to pass Soviet inspection but fail when in operation)
* Result: The largest non-nuclear explosion in the planet’s history

**Seven Testing Principles**



20



Testing can show that

**defects**

are

**present**



Cannot prove that there are

**no**

**defects**



Appropriate testing

**reduces**

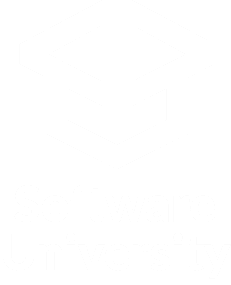
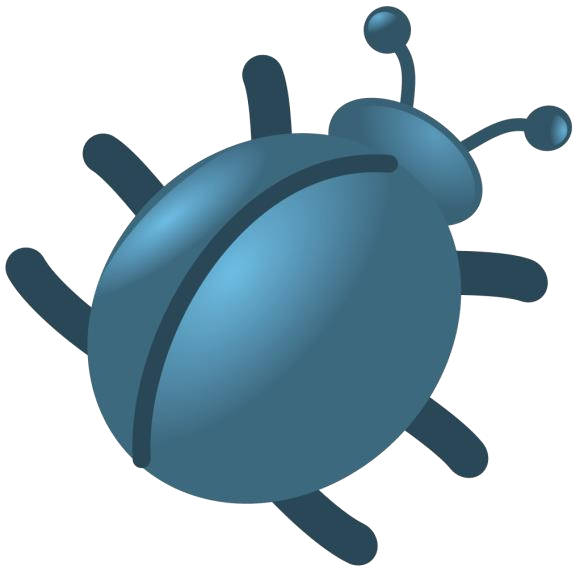
the

**probability**

for defects

**"Testing shows presence of**

**defects, not their absence"**



21



**All combinations**

of inputs and

preconditions are usually

almost

**infinite**

number



Testing everything is

**not**

**feasible**



Risk analysis and priorities

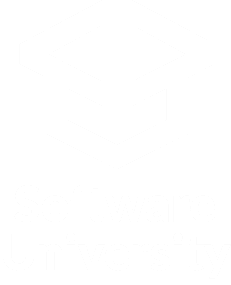
should be used to focus

testing efforts



**"Exhaustive testing is**

**impossible"**



22



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**

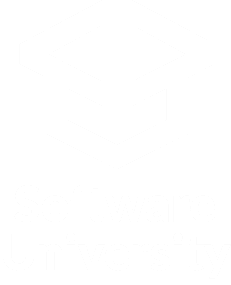
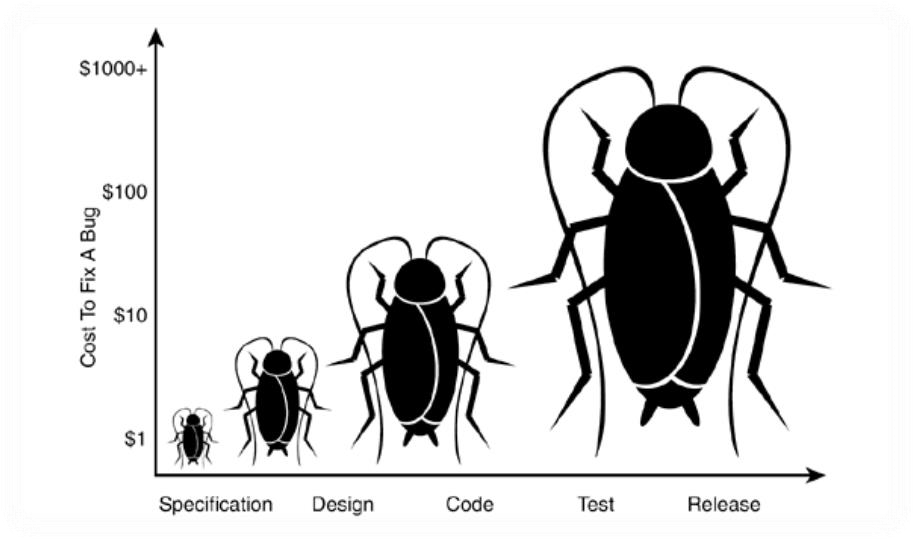
!

**"**

**Early testing saves time**

**and money**

**"**



23



Testing

**effort**

shall be focused

**proportionally**



To the expected and later

observed defect density of

modules



A

**small**

number of modules

u

~~s~~

ually

**contains**

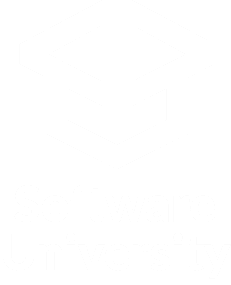
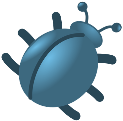
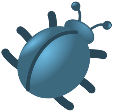
**most**

of the

**defects**

discovered

**"Defects cluster together"**



24



Same tests

**repeated**

over and

over again tend to

**lose**

their

**effectiveness**



**Previously**

**undetected**

defects remain

**undiscovered**



**New**

and

**modified**

test cases

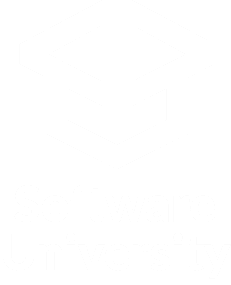
shou

d be developed

~~l~~

**"Beware of the pesticide**

**paradox"**



25



Testing is done

**differently**

in

different contexts



Safety

-

critical software is

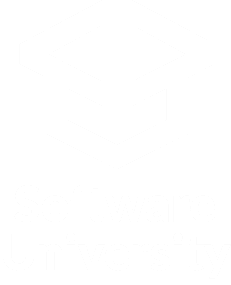
tested differently from

an e

-

commerce site

**"Testing is context dependent"**



26



Finding and fixing defects

itself does not help in these

cases:



The system built is

**unusable**



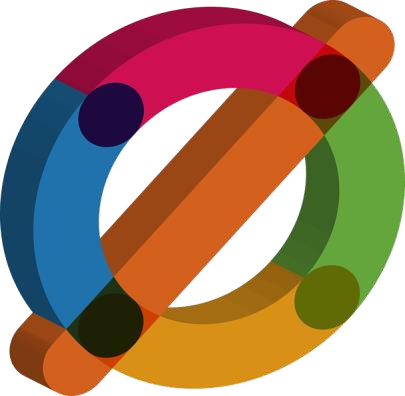
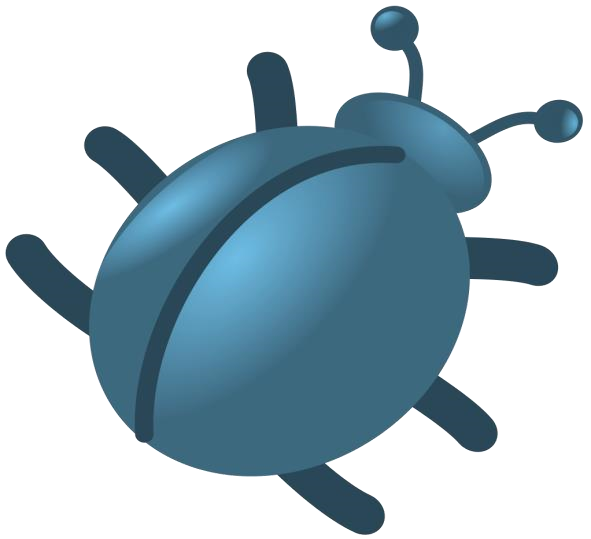
Does not fulfill the users’

**needs**

and

**expectations**

**"Absence of errors is a fallacy"**



**Test Process**



#### Test Activities and Tasks



**Test Process**

**Test**

**process**

**is a**

**set of test activities**

**which**

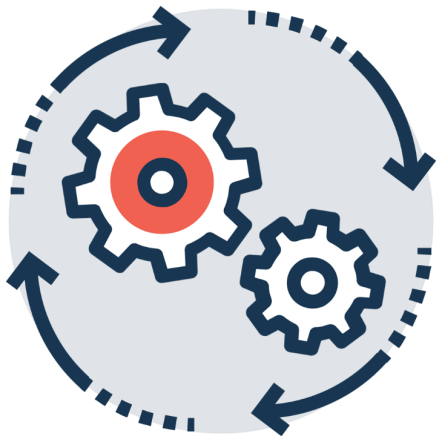
**depend**

**on many factors and the**

**main factor is the organization's**

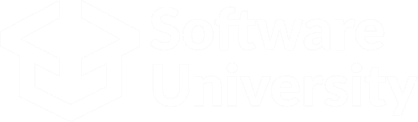
**test**

**strategy**



27

* The main groups of activities which are a part of the process



**Test Activities and Tasks**

29



**Begin**



**Test**

**Planning**



**Test**

**Analysis**

**and**

**Design**



**Test Implementation and Execution**



**Exit Criteria and Reporting**



**Test Closure Activities**



**End**



**Psychology of Testing**





Identifying defects may be perceived as

**criticism**



**Confirmation**

**bias**

can make it difficult to

**accept**

criticism



As a result, some people perceive

**testing**

**as**

a

**destructive**

activity



Good

**communicational**

**skill**

**s**

are a must

in order to

**avoid**

**conflict**

between

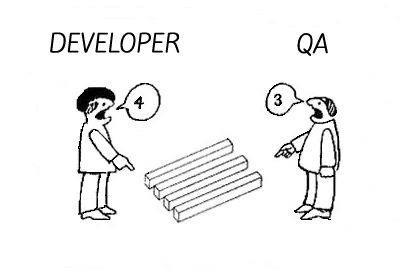
developers and QA

**Human Psychology**

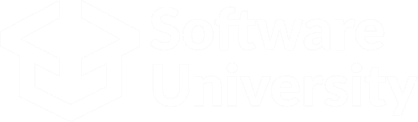


31

* **QA testers**
* Are perceived as destructive – only happy when they are finding faults
* Usually require good communication skills, tact & diplomacy

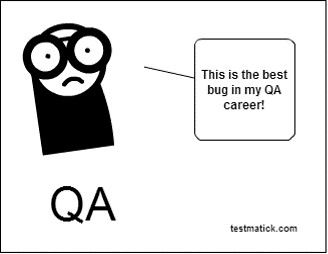
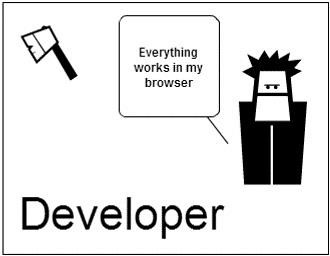


32

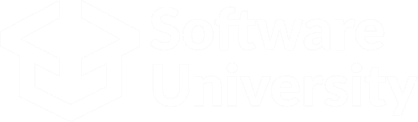


**QA vs Devs**

* Normally n~~e~~ed to be multi-talented (technical, testing, team skills)
* **Developers**
* Are perceived as very creative – they write code without which there would be no system



33

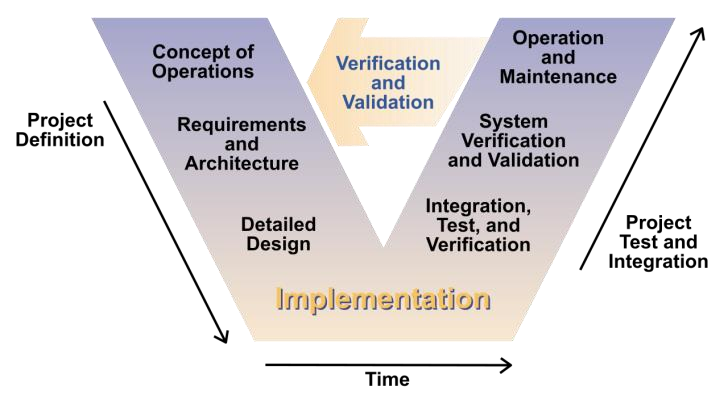


**Devs**

**vs QA**

* Are rarely good communicators
* Can often specialize in just one or two skills (VB, C++, JAVA, SQL)

**Software Development and Testing**



## Software Development Lifecycle Model



Describes the **stages** in the **development** of a **software**

**project**

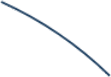
and the

**activities**

throughout each of them



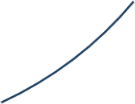
**Requirements**



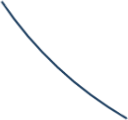
**Design**



**Implementation**



**Testing**



**Deployment**



**Maintenance**



## Software Testing Life Cycle



* Describes the **testing phases** and the **activities** throughout

the development of a software project



**Requirement**



**Planning**



**Analysis**



**Design**



**Implementation**



**Execution**



**Conclusion**



**What is a Development Methodology?** 

* A development **methodology** is a **set of practices** and procedures for **organizing** the software **development process**



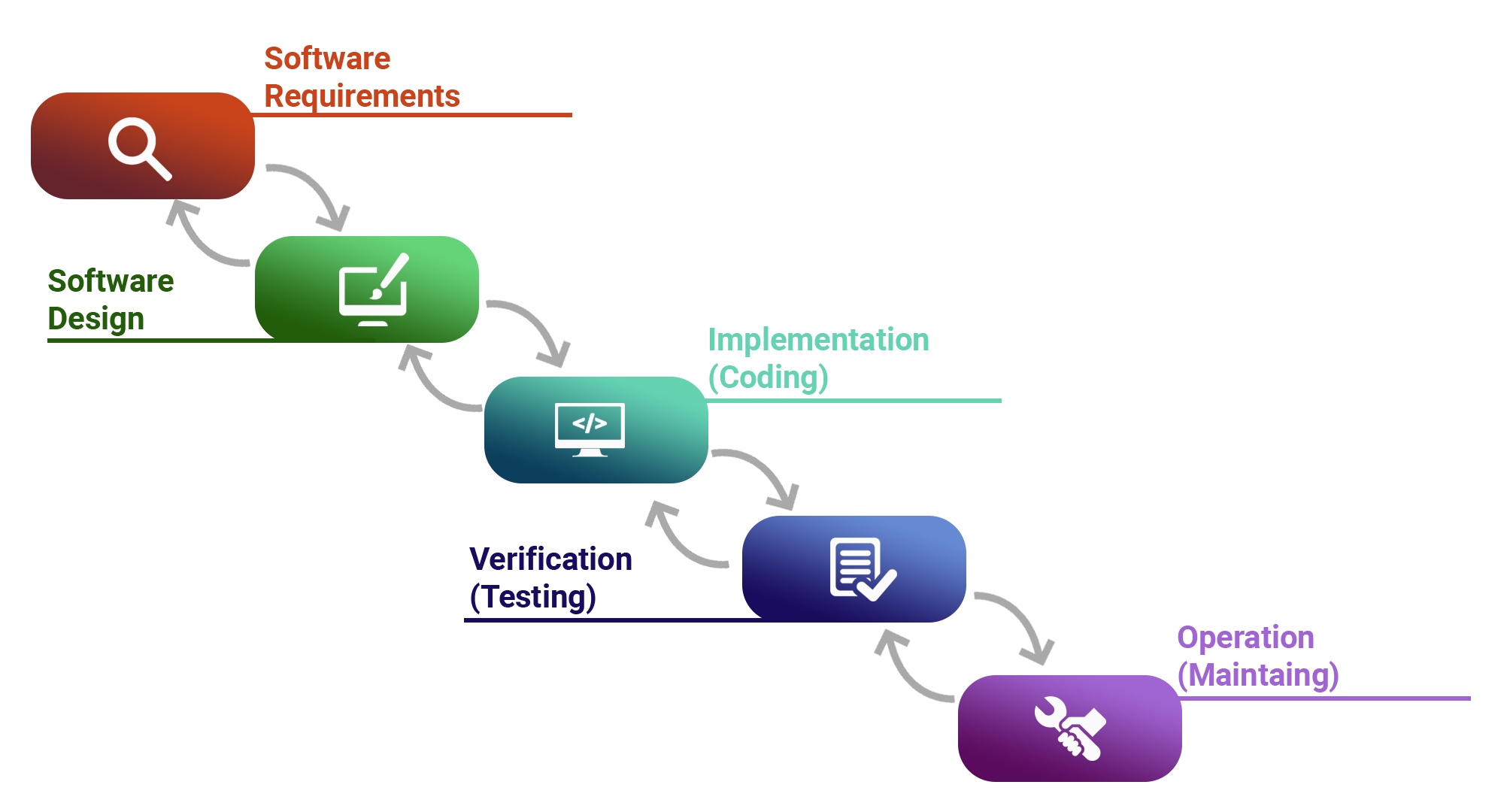
* Heavyweight and agile methodologies
* Heavy methodologies rely on formal procedures and documents
* Agile methodologies rely on small iterations and less formalities

### Development Methodologies

* The **"Waterfall"** Process
* Rational Unified Process (RUP)
* Microsoft Solutions Framework (MSF)
* **Agile** development processes
* Scrum, Kanban, Lean Development, Extreme Programming (XP), etc.



**The Waterfall Development Process**

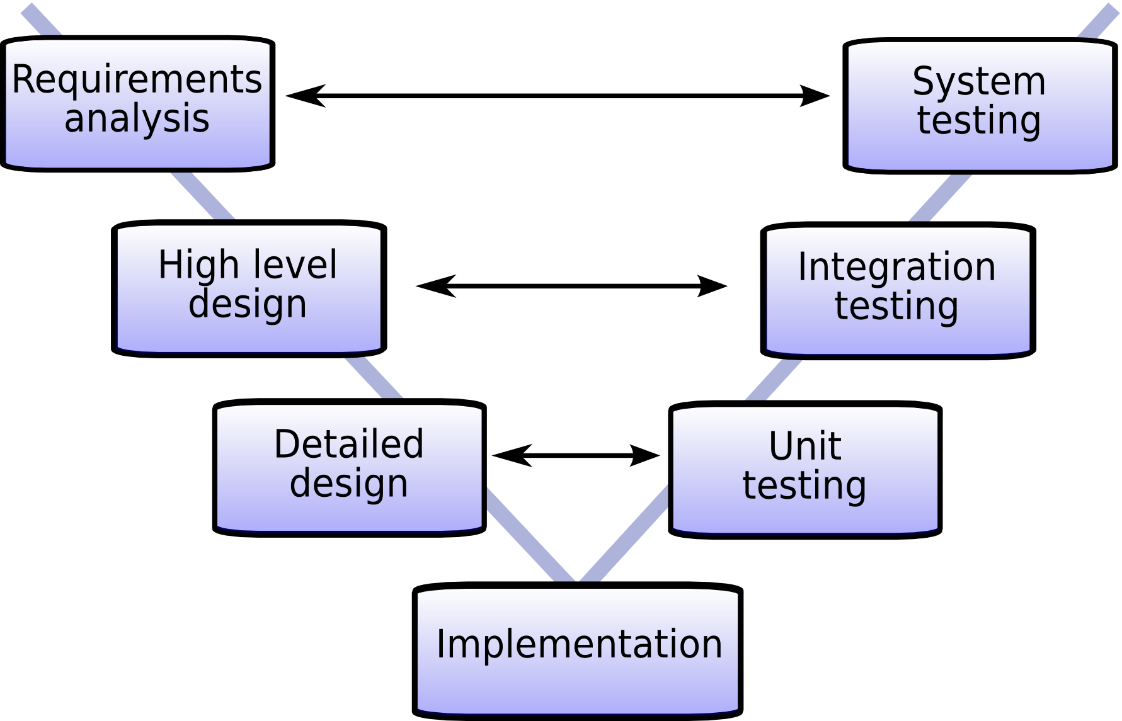


### The Waterfall Model

* The first fundamental model was the **"Waterfall"** model
* Impressively **simple** and very **well known**
* Only when one development level is completed will be initiated the next one
* Only between **adjacent levels** are there **feedback loops** and **revisions** of the previous level
* Emphasis is on **documentation**
* Testing is understood as a **"one time"** action at the end of the development

41

### V Model Development Process



#### V Model

* The model presents two branches
* **Development tasks**
* The process of design and implementation
* **Testing tasks**
* Verification and integration into bigger subsystems
* Both are corresponding activities of equal importance

**V Model Development Tasks**



**Functional**

system design



**Technical**

system design



**Component**

specification



**Programming**

/

**coding**

**Requirements' specification**

#### V Model Testing Tasks (1)

* **Component** (unit) **test**
* Verifies each software component
* Does it perform correctly according to its specification?
* **Integration test**
* Checks groups of components
* Do they collaborate correctly according to the technical system design?

#### V Model Testing Tasks (2)



Verifies the system as a whole



Does it meet the specified system requirements?



**Acceptance**

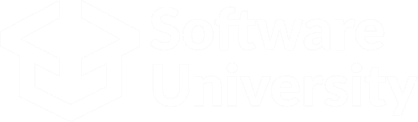
test



Does the system meet the customer requirements?

**System** test

Others



**Agile Methodologies**



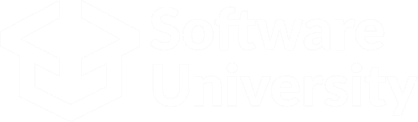
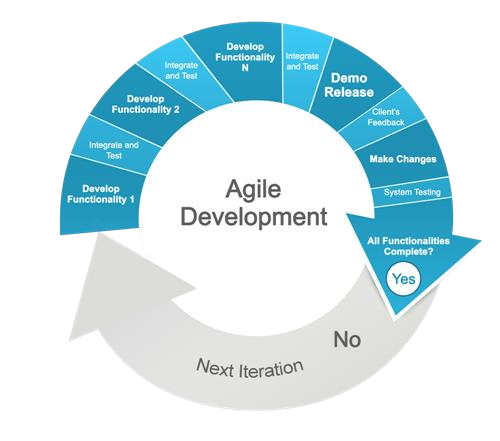
Scrum



Kanban



Lean Software Development



**Summary**

47



…



…



…





Definition of

**testing**



Importance



**Seven testing principles**



**Basic definitions**

of

**test process activities**



Communication between QA's and

Developers



**SDLC**

and

**STLC**



Most used

**development methodologies**

**Literature** ▪ QA Book:



48

[▪](http://www.istqb.org/downloads/foundation-level-documents)

[g](http://www.istqb.org/downloads/foundation-level-documents)

[www.istqb.or](http://www.istqb.org/downloads/foundation-level-documents)

▪

Tutorials:

[▪](http://www.w3schools.com/)

[www.w3schools.co](http://www.w3schools.com/)

[m](http://www.w3schools.com/)

▪

Selenium:

[▪](https://leanpub.com/selenium-webdriver-book)

[~~l~~](https://leanpub.com/selenium-webdriver-book)

[eanpub.com/seleniu](https://leanpub.com/selenium-webdriver-book)

[m](https://leanpub.com/selenium-webdriver-book)

[-](https://leanpub.com/selenium-webdriver-book)

[webdrive](https://leanpub.com/selenium-webdriver-book)

[r](https://leanpub.com/selenium-webdriver-book)

[-](https://leanpub.com/selenium-webdriver-book)

[k](https://leanpub.com/selenium-webdriver-book)

[boo](https://leanpub.com/selenium-webdriver-book)

▪

GitHub:

[▪](https://github.com/QualityAssuranceTeam/source)

[/](https://github.com/QualityAssuranceTeam/source)

[github.com](https://github.com/QualityAssuranceTeam/source)

[QualityAssuranceTea](https://github.com/QualityAssuranceTeam/source)

[m](https://github.com/QualityAssuranceTeam/source)

[e](https://github.com/QualityAssuranceTeam/source)

[/sourc](https://github.com/QualityAssuranceTeam/source)



© SoftUni

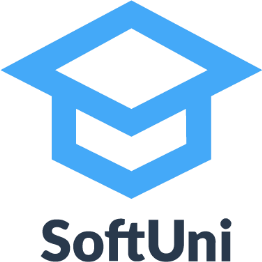
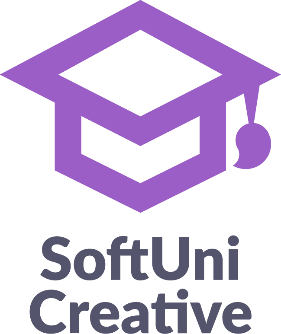
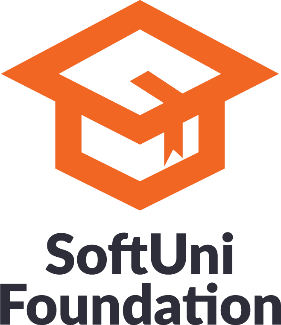
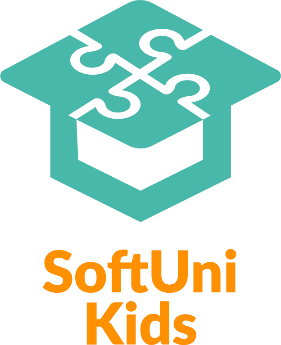
–

[https://about.softuni.or](https://softuni.org/)

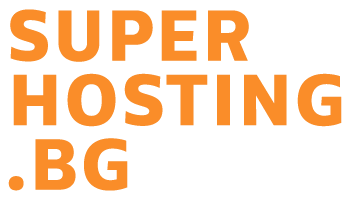
[g](https://softuni.org/)

[.](https://softuni.org/)

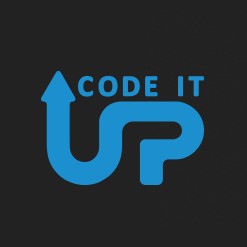
Copyrighted document. Unauthorized copy, reproduction or use is not permitted.



### SoftUni Diamond Partners



### Educational Partners



### License

* This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
* Unauthorized copy, reproduction or use is illegal
* © SoftUni – <https://about.softuni.bg/>
* © Software University – [https://softuni.bg](https://softuni.bg/) 

### Trainings @ Software University (SoftUni)





Software University

–

High

-

Quality Education,

Profession and Job for Software Developers

[](https://softuni.bg/)

[softuni.b](https://softuni.bg/)

[g](https://softuni.bg/)

[,](https://softuni.bg/)

[about.softuni.b](https://about.softuni.bg/)

[g](https://about.softuni.bg/)



Software University Foundation

[](https://softuni.foundation/)

[softuni.foundatio](https://softuni.foundation/)

[n](https://softuni.foundation/)



Software University @ Facebook

[](https://www.facebook.com/SoftwareUniversity)

[facebook.com/SoftwareUniversit](https://www.facebook.com/SoftwareUniversity)

[y](https://www.facebook.com/SoftwareUniversity)



Software University Forums

[](https://forum.softuni.bg/)

[forum.softuni.b](https://forum.softuni.bg/)

[g](https://forum.softuni.bg/)