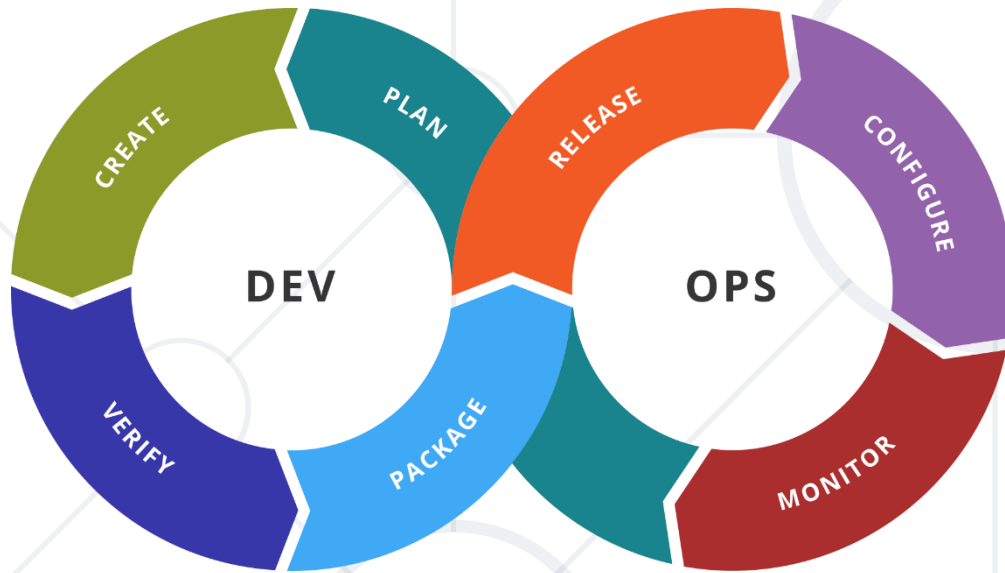


Puppet

Introduction and Basic Techniques



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://softuni.org>

You Have Questions?

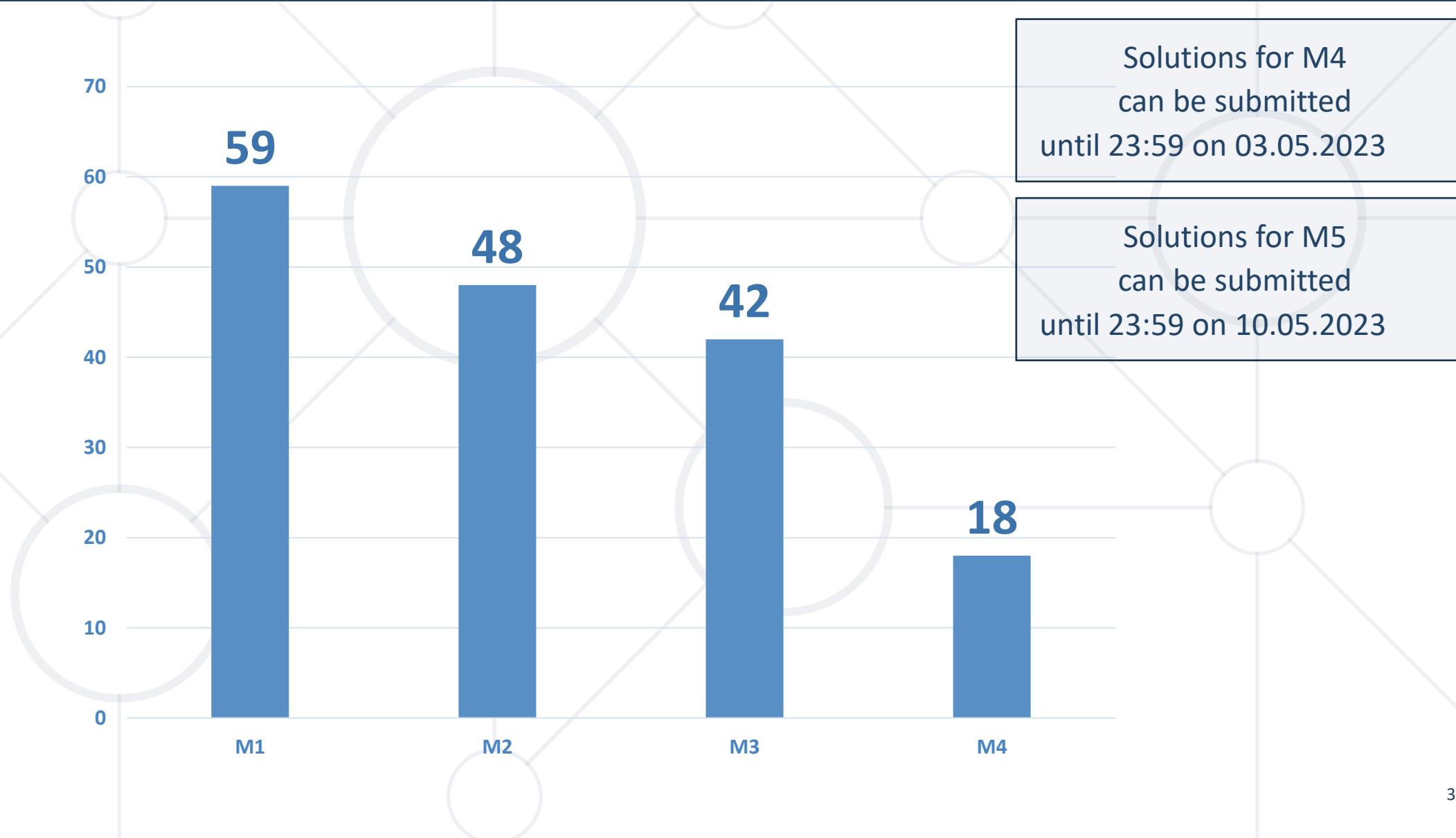
sli.do

#DevOps-23

facebook.com/groups

/DevOpsInfrastructureandConfigManagementApril2023

Homework Progress





Previous Module (M4)

Quick Overview

- Introduction to Chef
 - Components and architecture
 - Installation and first steps
- Working with Chef
 - Basic scenarios and files
 - Attributes, templates and files
- Advanced Chef
 - Custom resources and libraries
 - Testing



This Module (M5)

Topics and Lab Infrastructure

1. Introduction to Puppet
 - Components and architecture
 - Installation and first steps
2. Working with Puppet
 - Directories and environments
 - Facts and modules
3. Additional Puppet
 - Bolt
 - Integration with Vagrant



NETWORK: 192.168.99.0/24

.100

.101

.102

VM1
(server)

VM2
(client1)

VM3
(client2)

VirtualBox

Vagrant

Windows / Linux / macOS



Puppet 101

Introduction. Architecture. Installation

What is Puppet?

- Puppet manages and automates the configuration of servers
- Uses server-agent model
- Configurations are written in Puppet code (Ruby DSL)
- Masters can run on Linux/Unix systems
- Agents support Linux/Unix/Windows

- **Open Source Puppet**
 - Includes language, server, database, and agent
 - Works with bare metal, cloud and containers
- **Puppet Enterprise**
 - Web UI, additional integrations and modules
 - Support for virtual machines and additional capabilities
 - Reporting and role-based access control
 - Support

- **Puppet Server**

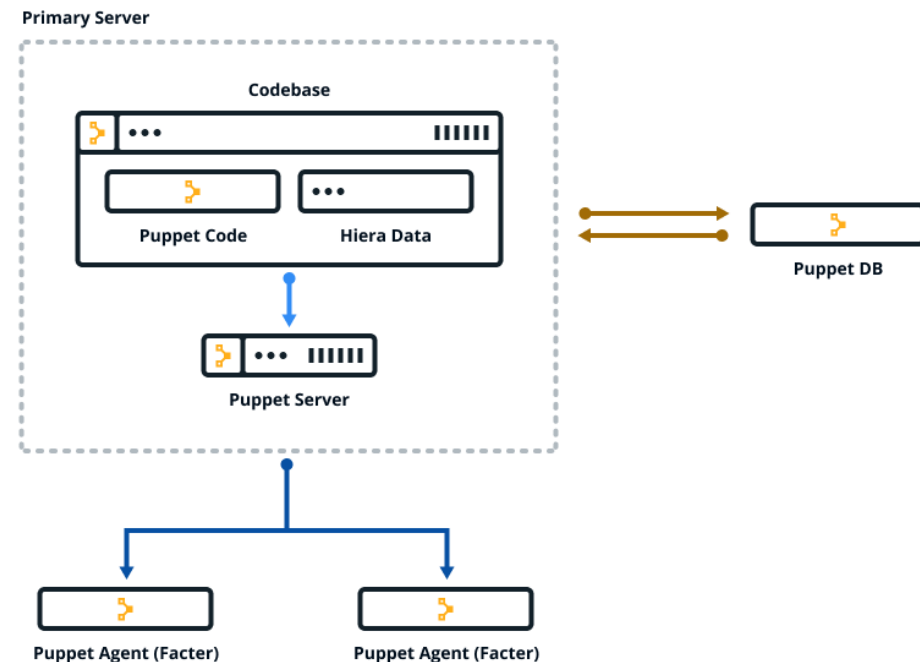
- Controls configuration for one or more managed nodes
- Communicate via HTTPS with the agents
- Has a built-in certificate authority
- Runs an agent to configure itself

- **Puppet Agent**

- Includes **Facter** for gathering information about a node
- And **Hiera** to separate the data from the code

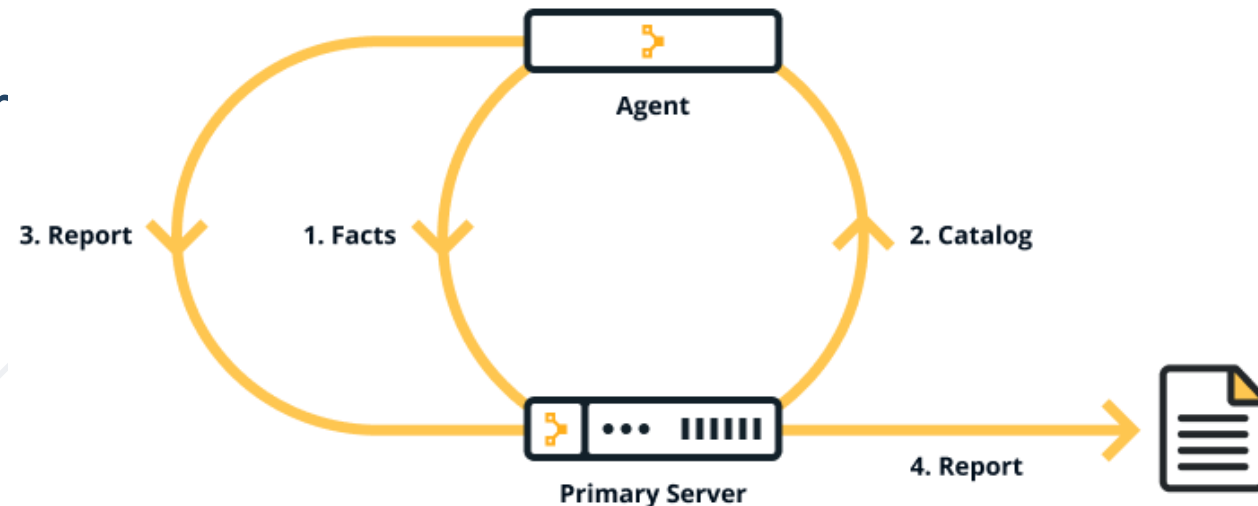
- **Puppet DB**

- Stores facts, catalog, reports, etc.



- **Puppet Forge**
 - Is a repository of existing modules
- **Puppet Development Kit (PDK)**
 - Used to create our own modules
- **VSCode Extension**
 - Puppet DSL extension for Windows, Linux, and macOS
- **Litmus**
 - Prepare and run acceptance tests over our code

- An **agent node** sends facts to the master and requests a catalog
- The **master compiles** and **returns** the node's catalog
- The **agent applies** the catalog to the node by checking each resource the catalog describes
- If the agent finds resources that are not in their desired state, it makes the changes necessary to correct them
- The agent reports back to the master



- **Resources** are the fundamental unit for modeling system configurations. Each resource describes some aspect of a system

```
<TYPE> { '<TITLE>':  
  <ATTRIBUTE> => <VALUE>,  
}
```

- **Manifests** are files that contain set of instructions to be executed
- **Classes** are code blocks that can be called in a code elsewhere
- **Modules** are collections of manifests and data (facts, files)



Practice: Puppet 101

Live Demonstration in Class



Puppet 102

Directories and Files. Environments. Modules

- **codedir**
 - Main directory for code and data
 - By default, it is at **/etc/puppetlabs/code**
 - For non-root users is at **~/.puppetlabs/etc/code**
 - Used by the **Puppet server** and **puppet apply**
- **confdir**
 - Contains configuration files and the SSL data
 - For the root user it is at **/etc/puppetlabs/puppet**
 - And for the non-root users it is at **~/.puppetlabs/etc/puppet**

- **Main manifest directory**
 - Catalog compilation starts with a single file or directory of manifests
 - The starting point is called **main manifest** or **site manifest**
 - Defaults to **<ENVIRONMENTS DIR>/<ENVIRONMENT>/manifests**
 - Controlled per environment with **environment.conf** file
 - Manifests in directory are treated as one and parsed in alphabetical order
- **The modulepath**
 - Global (**/etc/puppetlabs/code/modules**) and on environment level
 - Controlled per environment with **environment.conf** file

- Are branches that get turned into directories on the server
- Each has its own main manifest and module path
- We can use different versions of the same modules
- And use them against different group of nodes

- Per-node data that is available in Puppet manifest files as variables
- There are **core (built-in)**, **custom**, and **external** facts
- We can access them as **`$fact_name`** or **`$facts[fact_name]`**
- There are also legacy and modern facts
- Legacy facts are now hidden and are presented as structured ones
- For example, **architecture** is now **`os.architecture`**
- We can use the **facter** CLI to explore the facts

- Each module manages a specific task (create file, user, etc.)
- They are the basic building blocks
- Contain Puppet classes, defined types, tasks, task plans, functions, resource types and providers, and plugins (like custom types or facts)
- Installed in the Puppet modulepath
- We can install modules from Puppet Forge
- Or create our own modules



Practice: Puppet 102

Live Demonstration in Class



Puppet 103

Bolt. Vagrant Integration

- Open source orchestration tool
- Automates the manual work in infrastructure maintenance
- Suitable for on-demand tasks that are part of a workflow
- Doesn't require a separate server
- Connects directly via SSH or WinRM
- Doesn't need agents

- Files are organized in a repository
- There we store the inventory plus and additional files
- Desired state is expressed via YAML in plan files
- Each plan contains one or more tasks (built-in or custom)

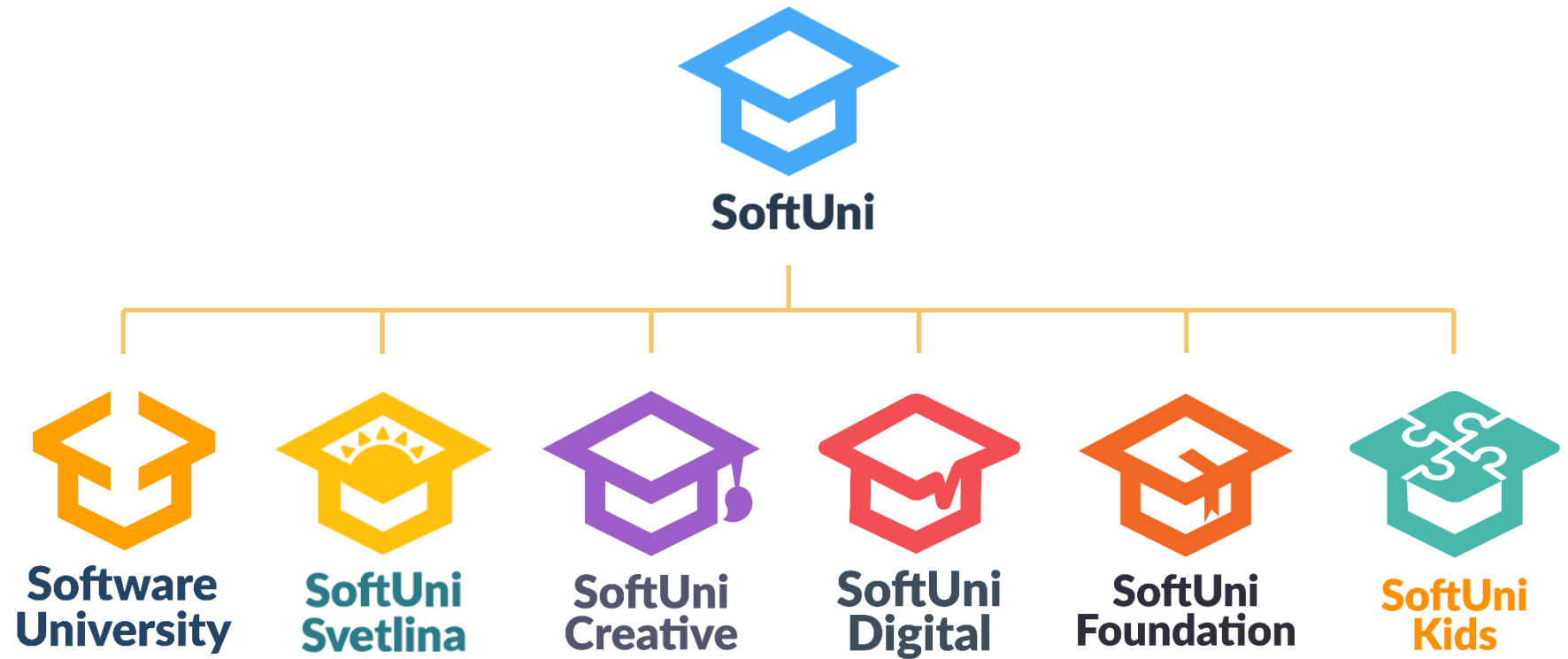
- There is a puppet (Puppet Apply) provisioner in Vagrant
- No master server is required
- Puppet should be installed on the target VM
- Custom manifest paths and files are supported
- We can select/set environment
- Custom modules and facts



Practice: Puppet 103

Live Demonstration in Class

Questions?



SoftUni Diamond Partners

SCHWARZ



Coca-Cola HBC
Bulgaria



Postbank

Решения за твоето утре



POKERSTARS



CAREERS



AMBITIONED

DXC
TECHNOLOGY



**SOFTWARE
GROUP**

Bosch.IO

INDEAVR
Serving the high achievers

 **DRAFT
KINGS**

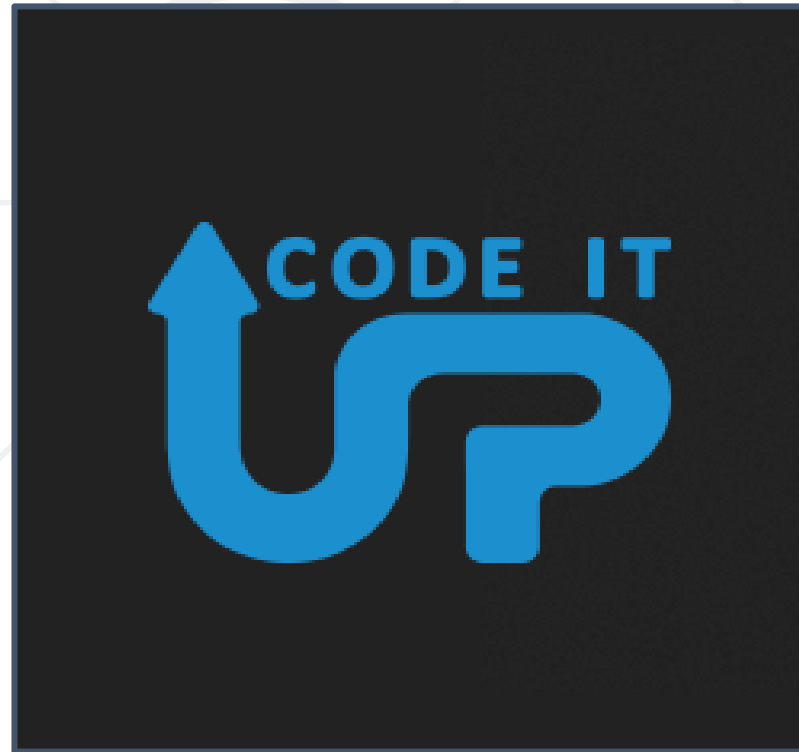
 **PHAR
VISION**



SmartIT

createX

**SUPER
HOSTING
.BG**



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



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg, softuni.org
- Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

