Git and GitHub

Version Control Systems

SoftUni Team

Technical Trainers

Software University

https://softuni.bg

Table of Contents

1. Software Configuration Management and **Source Control Systems**
   - Vocabulary: **Clone** a Repo, **Commit** a Changeset, **Push** the Changes, **Pull** Changes, **Merge** Changes
2. Introduction to **Git**
   - Working with git, Git Bash, and TortoiseGit
3. Introduction to **GitHub**
   - Create a Repo, Clone, Commit, Push, Conflicts

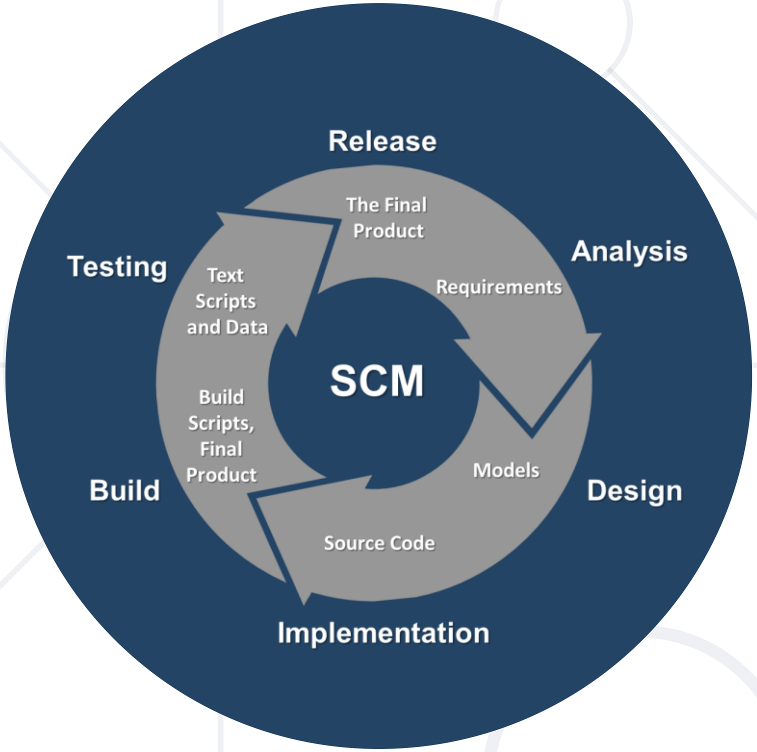
Have a Question?

sli.do

#fund-common

Source Control Systems: Lesson Summary

- **Source control systems** keep the source code (+ other project assets) in a shared **repository**
  - Developers can **clone** a repository, **pull** the latest version, **commit** & **push** local changes, view the change logs, etc.
- **Git** is the most popular source control system
  - Other version control systems: SVN, TFS, Perforce
- **GitHub** is the #1 site for Git project hosting
  - Git hosting + issue tracker + project tracker + build system



Software Configuration Management

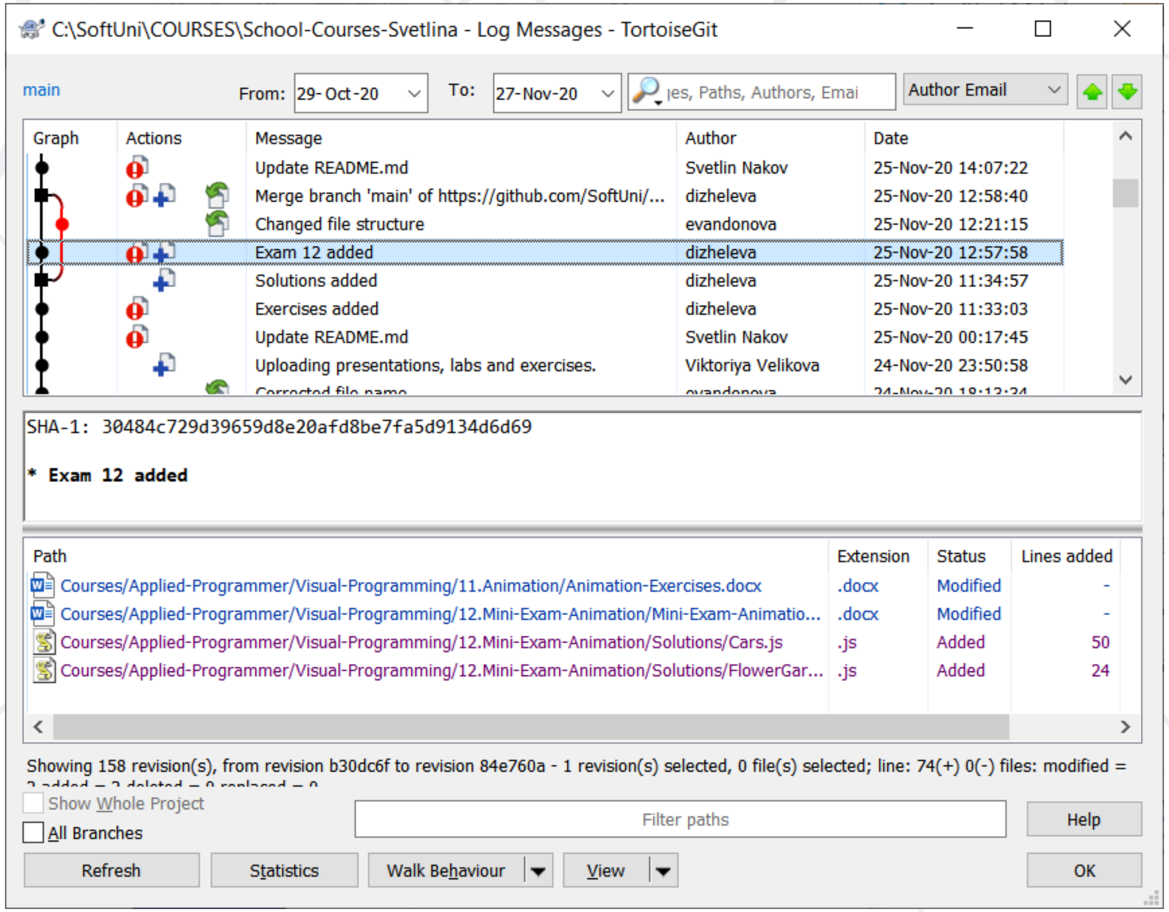
Working on Shared Code: Source Control Systems

Software Configuration Management

- **Version control** ≈ Software Configuration Management (SCM) ≈ **source control system**
  - A software engineering discipline
  - Consists of techniques, practices and tools for working on **shared source code** and files
  - Mechanisms for management, control and **tracking the changes**
  - Defines the process of **change management**
  - Keeps track of what is happening in the project over time
  - Solves **conflicts** in the changes

Change Log

- **Version control systems** keep their own **change log** (version history). It shows:
  - Who?
  - When?
  - Why?
  - What had been changed?
- Old versions could be **restored**



C:\SoftUni\COURSES\School-Courses-Svetlina - Log Messages - TortoiseGit

main From: 29-Oct-20 To: 27-Nov-20 ıes, Paths, Authors, Emai Author Email

Graph	Actions	Message	Author	Date
		Update README.md	Svetlin Nakov	25-Nov-20 14:07:22
		Merge branch 'main' of https://github.com/SoftUni/...	dizheleva	25-Nov-20 12:58:40
		Changed file structure	evandonova	25-Nov-20 12:21:15
		Exam 12 added	dizheleva	25-Nov-20 12:57:58
		Solutions added	dizheleva	25-Nov-20 11:34:57
		Exercises added	dizheleva	25-Nov-20 11:33:03
		Update README.md	Svetlin Nakov	25-Nov-20 00:17:45
		Uploading presentations, labs and exercises.	Viktoriya Velikova	24-Nov-20 23:50:58
		Corrected file name	evandonova	24-Nov-20 18:13:34

```
SHA-1: 30484c729d39659d8e20afd8be7fa5d9134d6d69

* Exam 12 added
```

Path	Extension	Status	Lines added
Courses/Applied-Programmer/Visual-Programming/11.Animation/Animation-Exercises.docx	.docx	Modified	-
Courses/Applied-Programmer/Visual-Programming/12.Mini-Exam-Animation/Mini-Exam-Animatio...	.docx	Modified	-
Courses/Applied-Programmer/Visual-Programming/12.Mini-Exam-Animation/Solutions/Cars.js	.js	Added	50
Courses/Applied-Programmer/Visual-Programming/12.Mini-Exam-Animation/Solutions/FlowerGar...	.js	Added	24

Showing 158 revision(s), from revision b30dc6f to revision 84e760a - 1 revision(s) selected, 0 file(s) selected; line: 74(+) 0(-) files: modified = 2 added = 2 deleted = 0 replaced = 0

□ Show Whole Project

□ All Branches

Vocabulary: Repository (Repo)

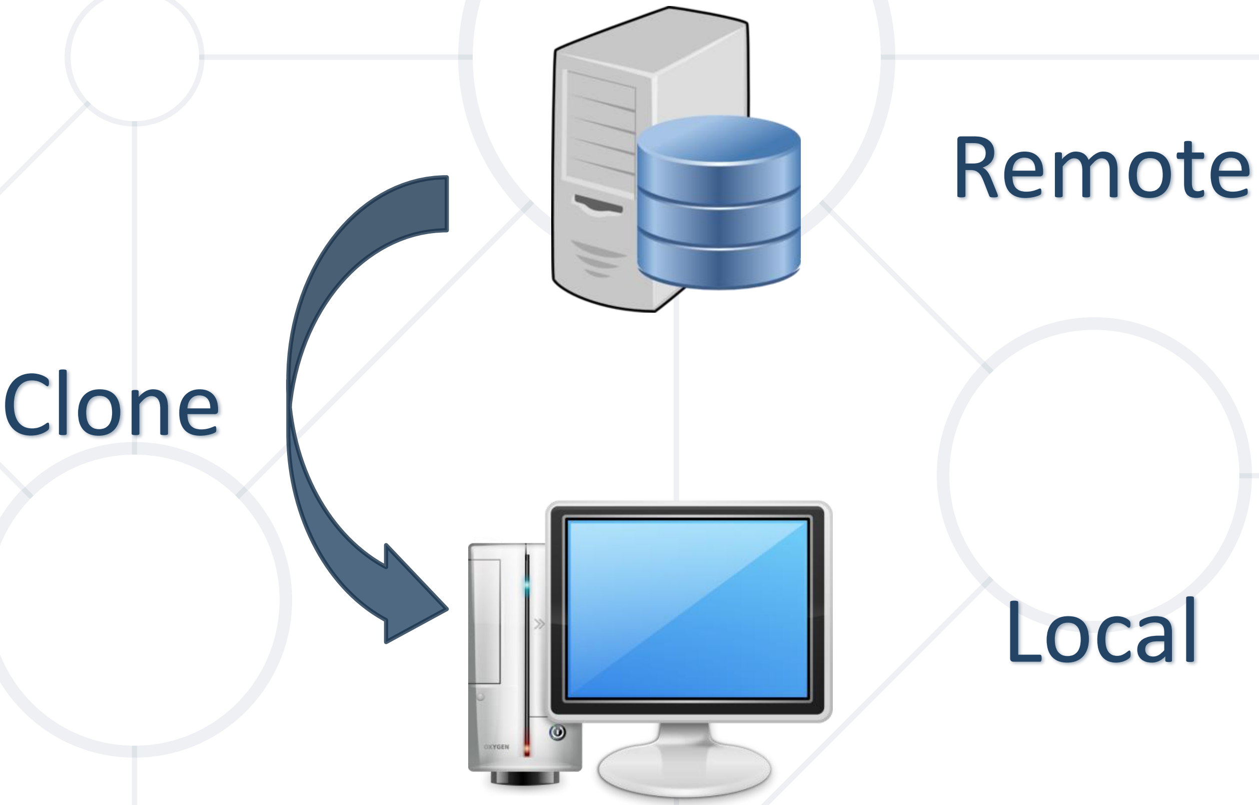
- **Repo** holds the project in a remote server

Remote

Local

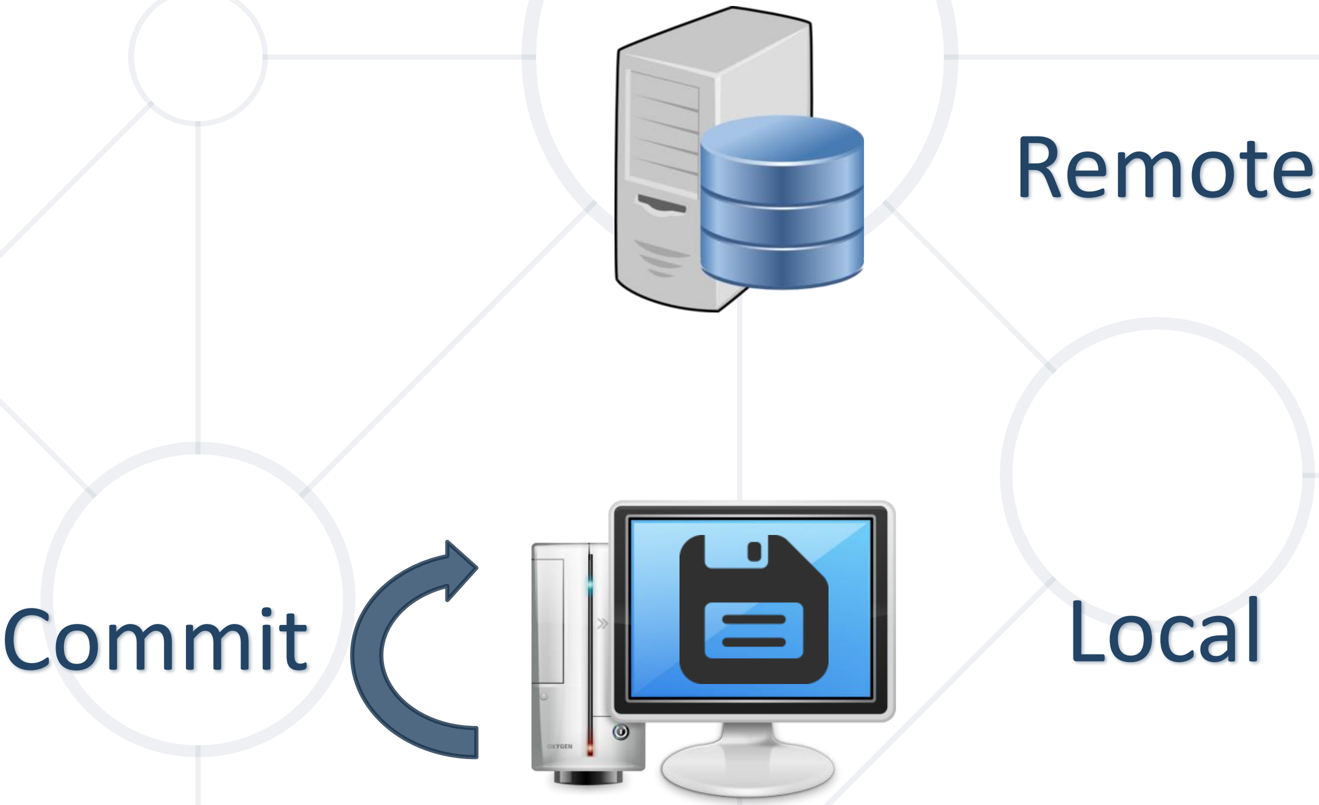
Vocabulary: Clone

- **Clone** == download a **local copy** of the remote project



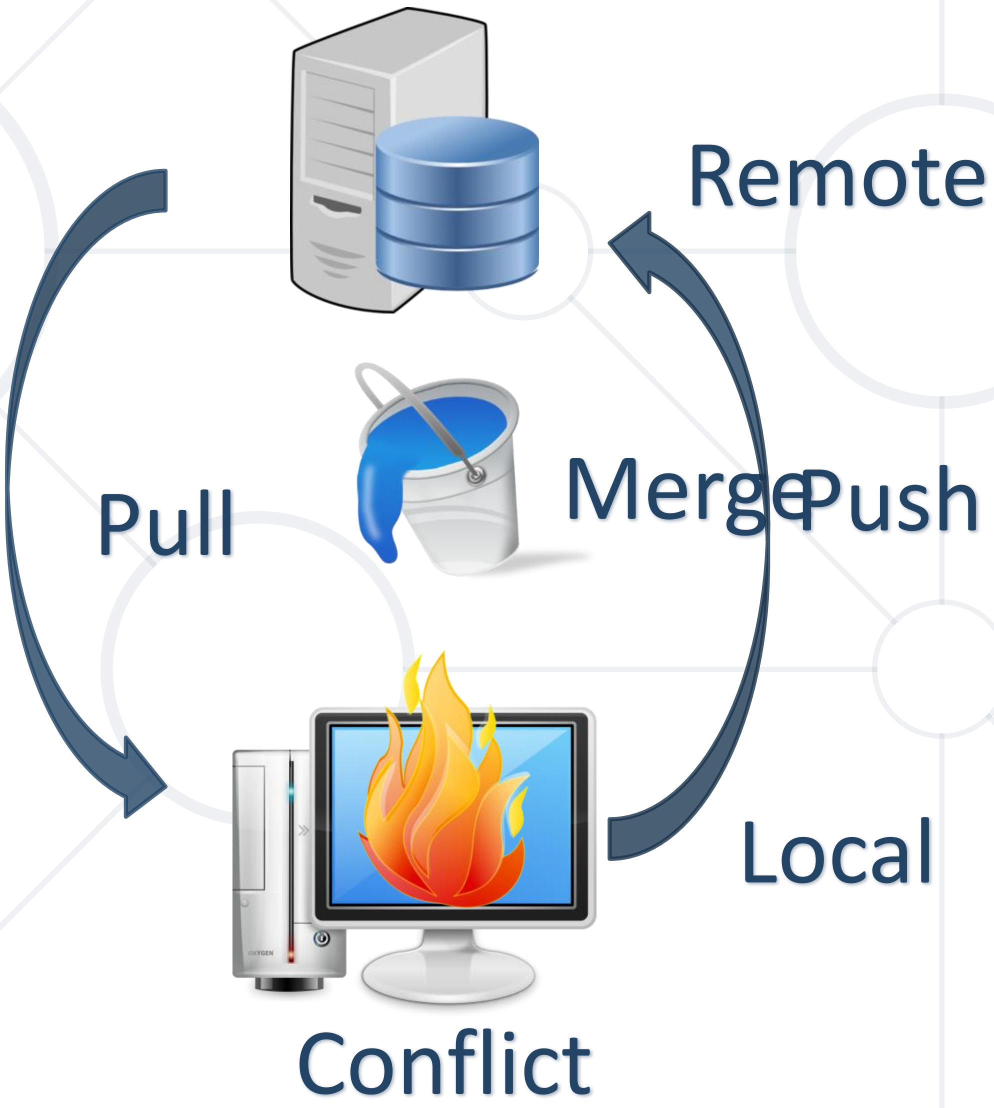
Vocabulary: Commit

- **Commit** == saves a set of changes locally

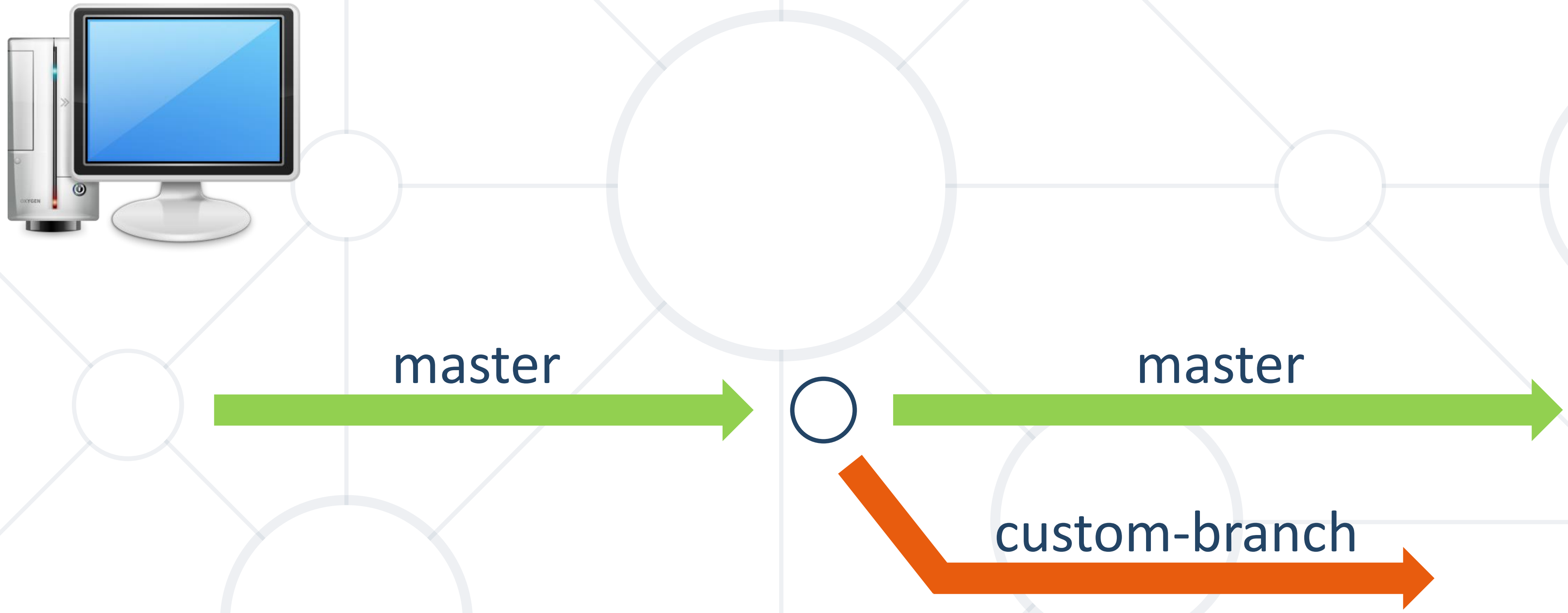


Vocabulary: Sync (Pull / Push)

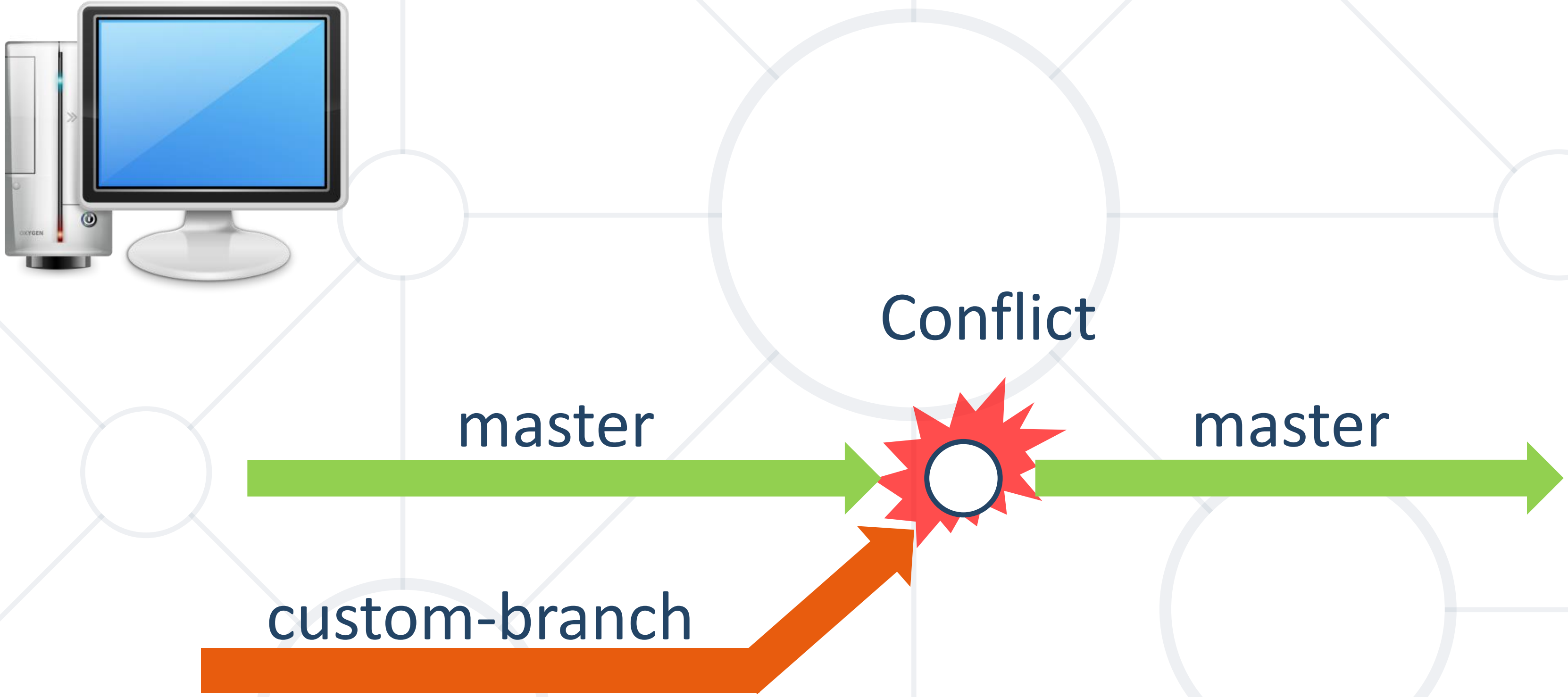
- **Pull** – **take** and **merge** the changes from the Remote
- **Push** – **send** local changes to the Remote



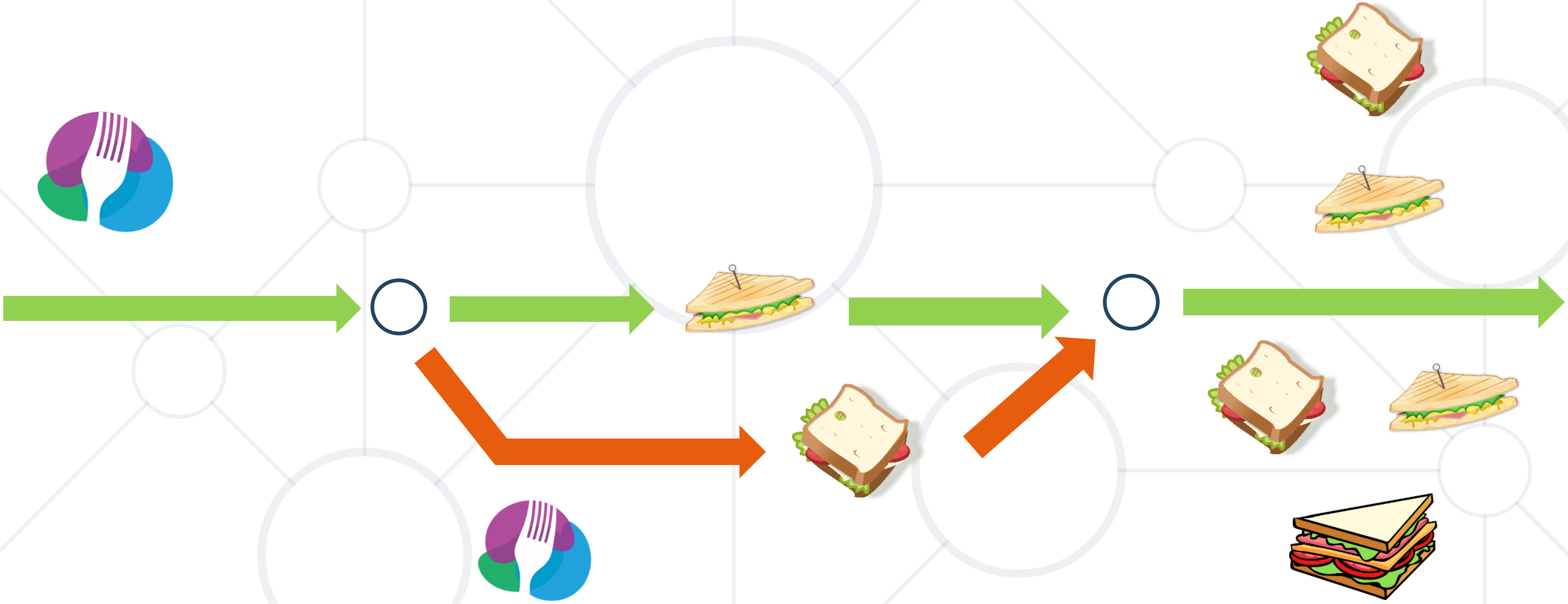
Vocabulary: Branch



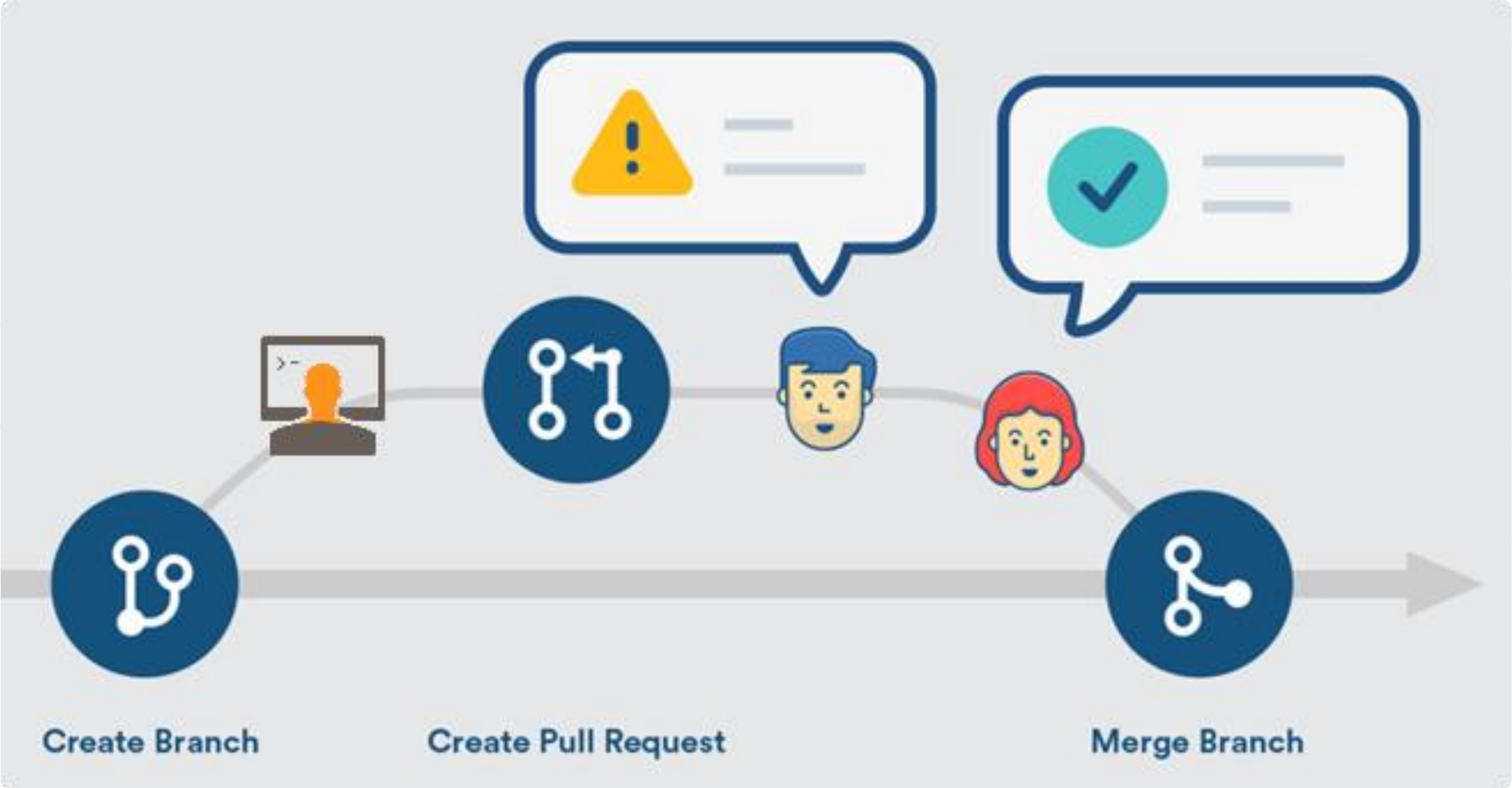
Vocabulary: Merge Branches



Example: Branches



Pull Requests: The Code Review Process



Git

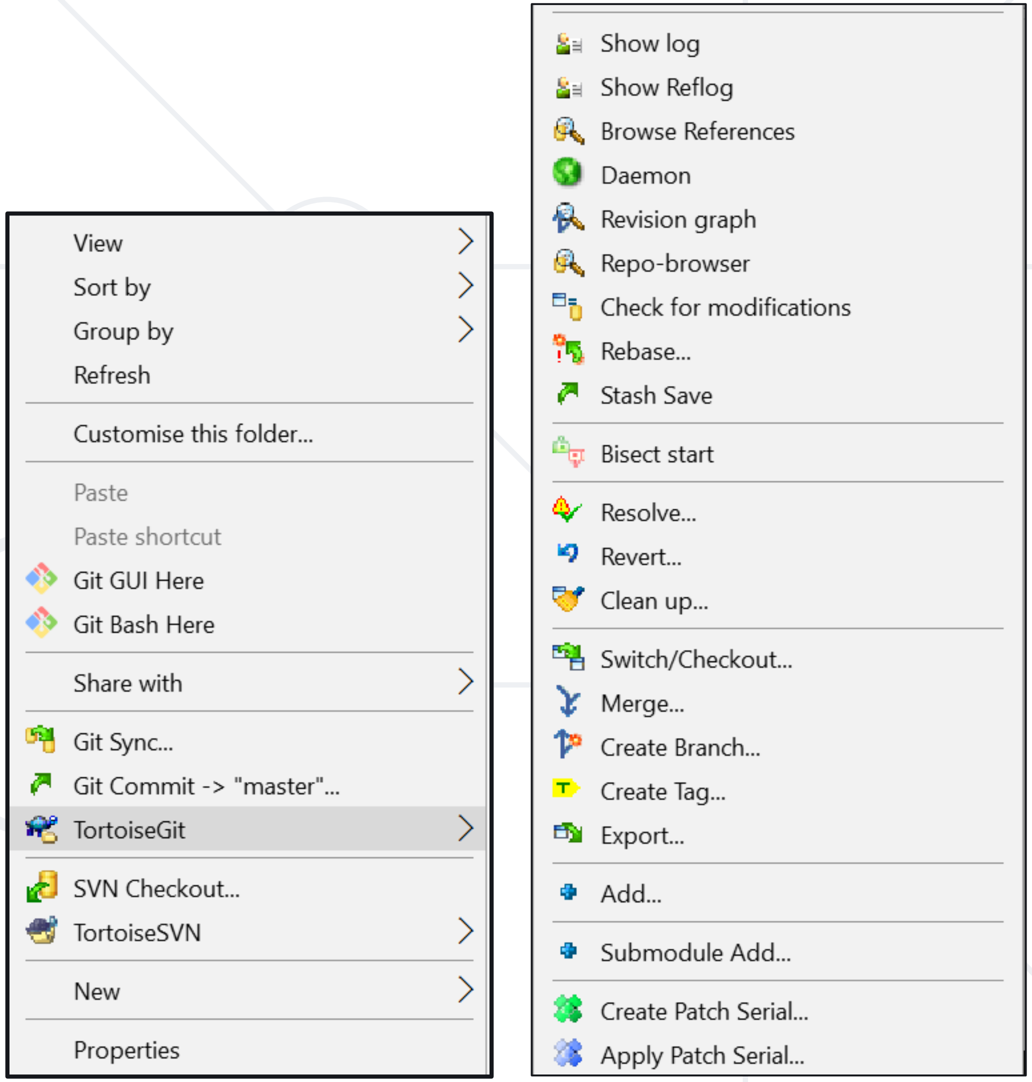
World's #1 Source Control System

What is Git?

- **Git** == distributed **source-control system**
  - The most popular in the world
  - Free, open-source software
- Works with **local** and **remote** repositories
- **Git bash** – command line interface for Git
- Runs on Linux, macOS and Windows (**msysGit**)
  - https://git-scm.com

Using Git

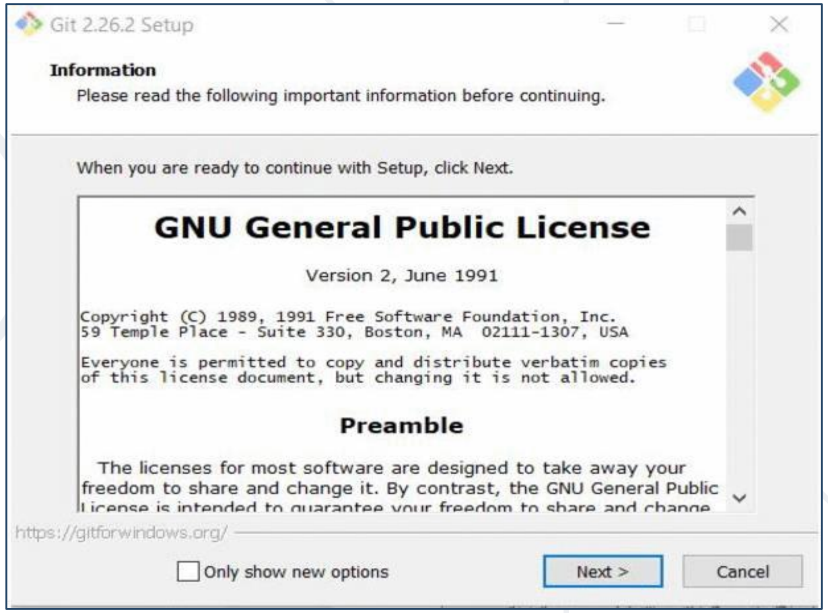
- Console-based Git client
  - **git, Git Bash**
- Windows GUI client – **TortoiseGit**
  - https://tortoisegit.org/download
- Visual Studio / Eclipse plug-ins
- **GitHub Desktop** client
  - https://desktop.github.com



Installing Git

- Git installation on Windows: Git for Windows (msysGit)
  - https://git-scm.com/downloads
  - Options to select (they should be selected by default)
    - "Use Git Bash Only"
    - "Checkout Windows-style, Commit Unix-style Endings"
- Git installation on Linux:

```
sudo apt-get install git
```



Git 2.26.2 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

Basic Git Commands (1)

- Cloning an existing Git repository

```
git clone [remote url]
```

- Fetch and merge the latest changes from the remote repository

```
git pull
```

- Preparing (adding / selecting) files for a commit

```
git add [filename] ("git add ." adds everything)
```

- Committing to the local repository

```
git commit –m "[your message here]"
```

Basic Git Commands (2)

- Check the status of your local repository (see the local changes)

```
git status
```

- Creating a new local repository (in the current directory)

```
git init
```

- Creating a remote (assign a short name for remote Git URL)

```
git remote add [remote name] [remote url]
```

- Pushing to a remote (send changes to the remote repository)

```
git push [remote name] [local name]
```

Git: Live Demo

Checkout → Modify → Commit → Push

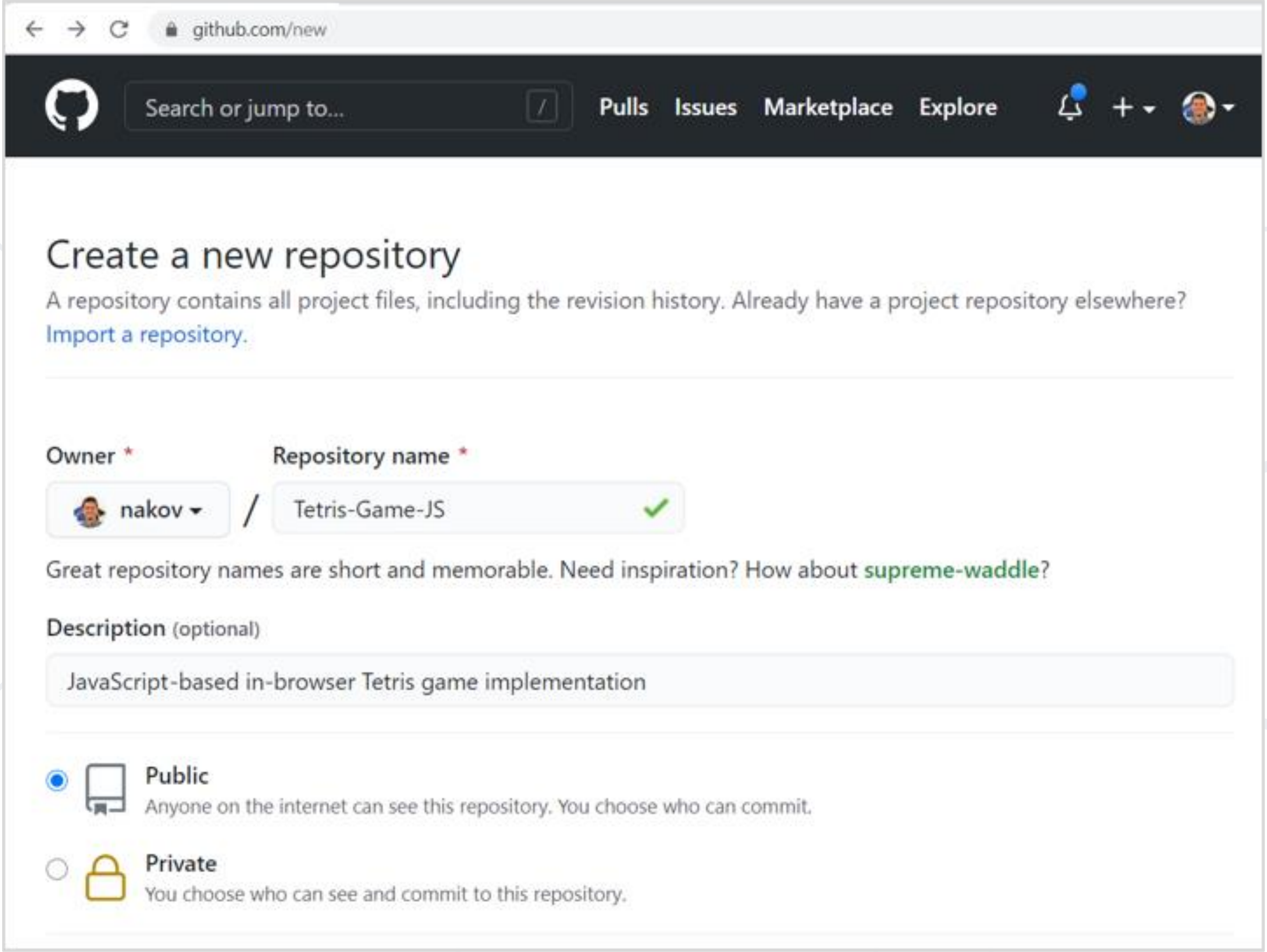
GitHub

The World's #1 Developer Collaboration Cloud

What is GitHub?

- **GitHub** is the world's #1 source code hosting site
  - Free for open-source projects
  - Paid plans for private repositories
- GitHub provides:
  - Git **source code** repository
  - **Issue tracker** (bug tracker)
  - **Project board** (Kanban style)
  - **Wiki** pages (documentation)
  - **Code reviews** (pull requests)
  - **Build system** (actions)
  - **Site hosting** (pages)
  - **Discussions** (forum)

Creating a GitHub Repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * nakov / Repository name * Tetris-Game-JS

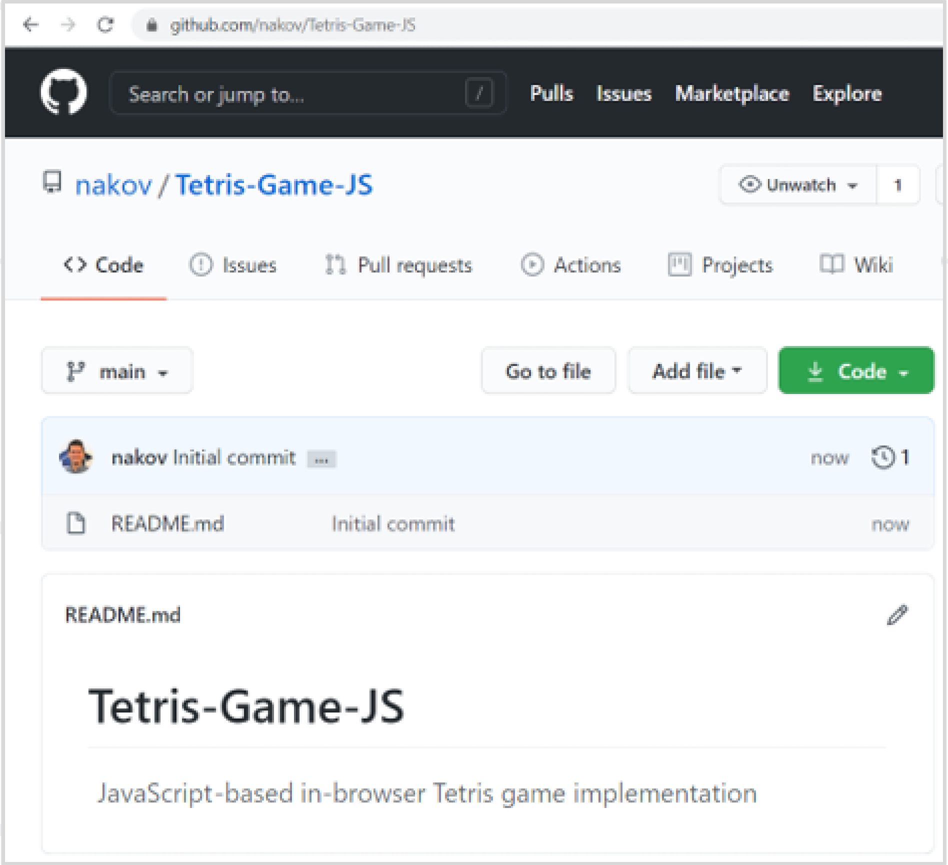
Great repository names are short and memorable. Need inspiration? How about supreme-waddle?

Description (optional)

JavaScript-based in-browser Tetris game implementation

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.



nakov / Tetris-Game-JS

nakov Initial commit ... now 1

README.md Initial commit now

README.md

Tetris-Game-JS

JavaScript-based in-browser Tetris game implementation

GitHub – Example

- **Clone** a repository from GitHub

```
git clone https://github.com/SoftUni/playground
```

- **Modify** local files

```
notepad README.md
```

- **Commit** changes (local)

```
git add . & git commit -m "Added something"
```

- **Push** the changes to GitHub

```
git push
```

GitHub: Live Demo

Create Repo → Edit Files → Checkout → Push

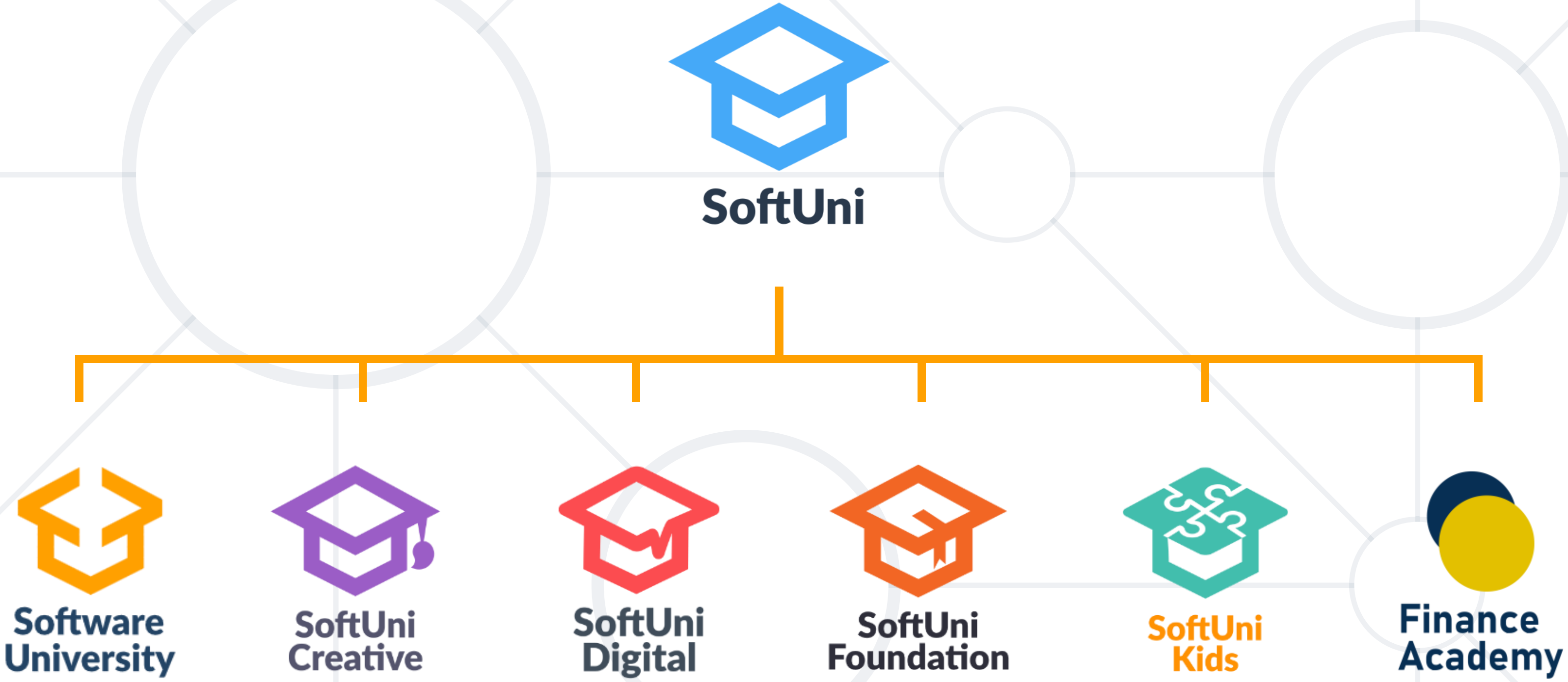
Live Exercises

Creating a Repo, Cloning a Repo, Commit and Push Changes, Resolve Conflicts, Team Interactions

Summary

- Use **version control systems** to work in a team
  - Keep the shared code in a central repository
  - Handle merge conflicts with ease
- Important **Git** commands:
  - **`clone`**, **`add`**, **`commit`**, **`pull`**, **`push`**
- **GitHub** == the world's most used software project hosting platform
  - Git repository, issue tracker, Kanban board, Wiki

Questions?



SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

Finance Academy

SoftUni Diamond Partners

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

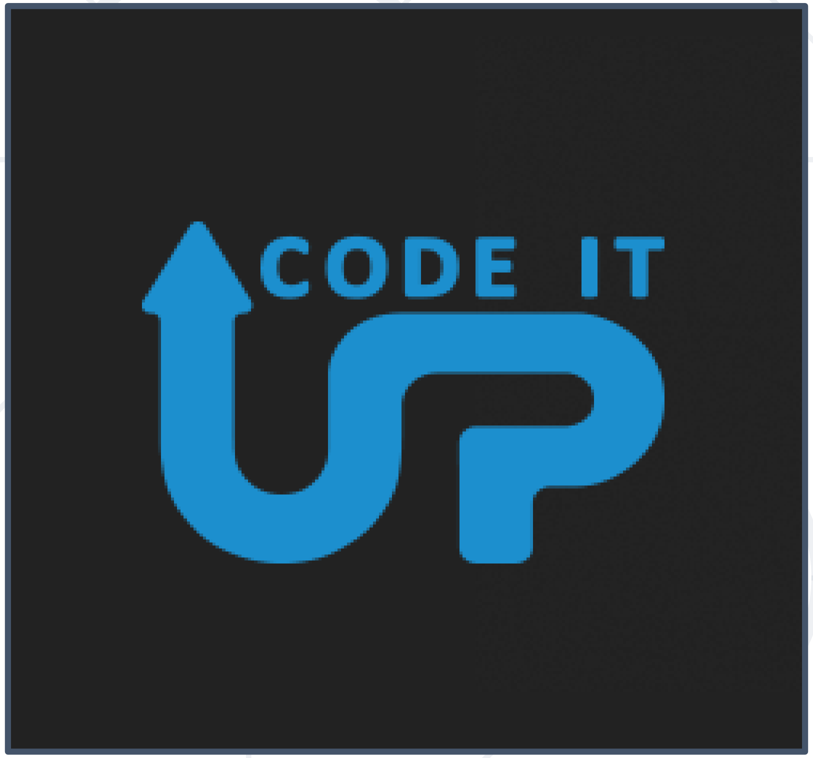
Postbank
Решения за твоето утре

BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners



Trainings @ Software University (SoftUni)

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

License