WELCOME

TechPro

[Pavlos Xouplidis]

{Advanced Git and Github}

[14/10/2024]

TechPro

In this Course

1. Understand the use of GitHub and be familiar with the UI
2. Create a GitHub account, clone and fork your first code in VS Code
3. Understand the most common Branching Strategies
4. You are familiar with developer's collaboration with Git and GitHub
5. Work on a git repository with others and request your first pull

TechPro

Assesment

{20 minutes assigment}

TechPro

Discussion

{Do you know GitHub?
What is the difference with
Git?}

TechPro

What is GitHub ?

GitHub is a cloud-based platform where you can store, share, and work together with others to write code.

TechPro

The story of GitHub

The GitHub service was developed by Chris Wanstrath, P. J. Hyett, Tom Preston-Werner, and Scott Chacon using Ruby on Rails, and started in February 2008.

On February 24, 2009, GitHub announced that within the first year of being online, GitHub had accumulated over 46,000 public repositories

On June 4, 2018, Microsoft announced its intent to acquire GitHub for US$7.5 billion

TechPro

The logo of GitHub

GitHub's mascot is an anthropomorphized "octocat" with octopus-like arm

https://octodex.github.com/

Competitors of GitHub

Bitbucket

SOURCEFORGE

GitLab

Key features of GitHub

- **Version control:** Tracks changes to your code over time, making it easy to revert to previous versions or compare different iterations.
- **Collaboration:** Allows multiple developers to work on the same project simultaneously, merging their changes and resolving conflicts.
- **Open-source development:** A vast community of developers contributes to open-source projects hosted on GitHub, fostering collaboration and innovation.

TechPro

Key features of GitHub

- **Issue tracking:** Helps manage bugs, feature requests, and other tasks related to a project.
- **Project management:** Provides tools for planning, organizing, and tracking the progress of projects.

TechPro

Why use GitHub?

- **Collaboration:** Work with others on projects, no matter where they are located.
- **Backup:** Keep your code safe and accessible.
- **Version control:** Easily track changes to your code.
- **Open-source community:** Connect with other developers and contribute to open-source projects.

TechPro

GitHub Client Apps

GitHub CLI is an open-source tool for using GitHub from your computer's command line.

GitHub Desktop, you can interact with GitHub using a GUI instead of the command line or a web browser.

TechPro

Collaborative Coding

Repositories
A repository is the most basic element of GitHub. It's a place where you can store your code, your files, and each file's revision history. Repositories can have multiple collaborators and can be either public or private.

TechPro

Collaborative Coding

Fork
A fork is a new repository that shares code and visibility settings with the original "upstream" repository. Forks are often used to iterate on ideas or changes before they are proposed back to the upstream repository, such as in open-source projects or when a user does not have write access to the upstream repository.

TechPro

Collaborative Coding

You can use forks to propose changes related to fixing a bug.
Rather than logging an issue for a bug you have found, you can:

- ➢ Fork the repository.
- ➢ Make the fix.
- ➢ Submit a pull request to the project owner.

TechPro

Collaborative Coding

Pull request
A pull request is a proposal to merge a set of changes from one branch into another. In a pull request, collaborators can review and discuss the proposed set of changes before they integrate the changes into the main codebase. Pull requests display the differences, or diffs, between the content in the source branch and the content in the target branch.

Collaborative Coding

Codespaces
A codespace is a development environment that's hosted in the cloud. The codespace development environment is created from an Ubuntu Linux image that includes a selection of popular languages and tools

TechPro

Collaborative Coding

GitHub Discussions
GitHub Discussions is a collaborative communication forum for the community around an open source or internal project. Repository owners and people with write access can enable GitHub Discussions for a community on their public and private repositories.

TechPro

Collaborative Coding

GitHub Copilot
GitHub Copilot is an AI coding assistant that helps you write code faster and with less effort, allowing you to focus more energy on problem solving and collaboration.

TechPro

Project Management

Projects
A project is an adaptable collection of items that you can view as a table, a kanban board, or a roadmap and that stays up-to-date with GitHub data. Your projects can track issues, pull requests, and ideas that you note down.

Project Management

Issues:
You can create issues in your repository to plan, discuss, and track work. Issues can track bug reports, new features and ideas, and anything else you need to write down or discuss with your team.

TechPro

Graphic user Interface for Git

https://git-scm.com/downloads/guis

TechPro

What is Branching Strategy?

- **Branching** allows teams of developers to easily collaborate inside of one central code base. When a developer creates a branch, the version control system creates a copy of the code base at that point in time.

- **Branching models** often differ between teams and are the subject of much debate in the software community.

TechPro

Choosing the right Branching strategy

The best branching strategy for your team depends on factors such as project size, team size, development process, and risk tolerance. Consider the following questions:

- ✓ How often do you release new versions?
- ✓ How complex are your features?
- ✓ What is your team's experience level with version control?
- ✓ What is your risk tolerance for breaking changes?

TechPro

Branching strategies

- Trunk-Based Development (TBD)
- Release branching
- Feature branching (hotfix and Task)

TechPro

➢Trunk-Based Development (TBD)

Core principle: All developers work on a single branch, the "trunk" or "main" branch.

Benefits: Simplicity, faster releases, and reduced merge conflicts.

Challenges: Higher risk of breaking changes, difficulty managing long-running features.

TechPro

Release Branching

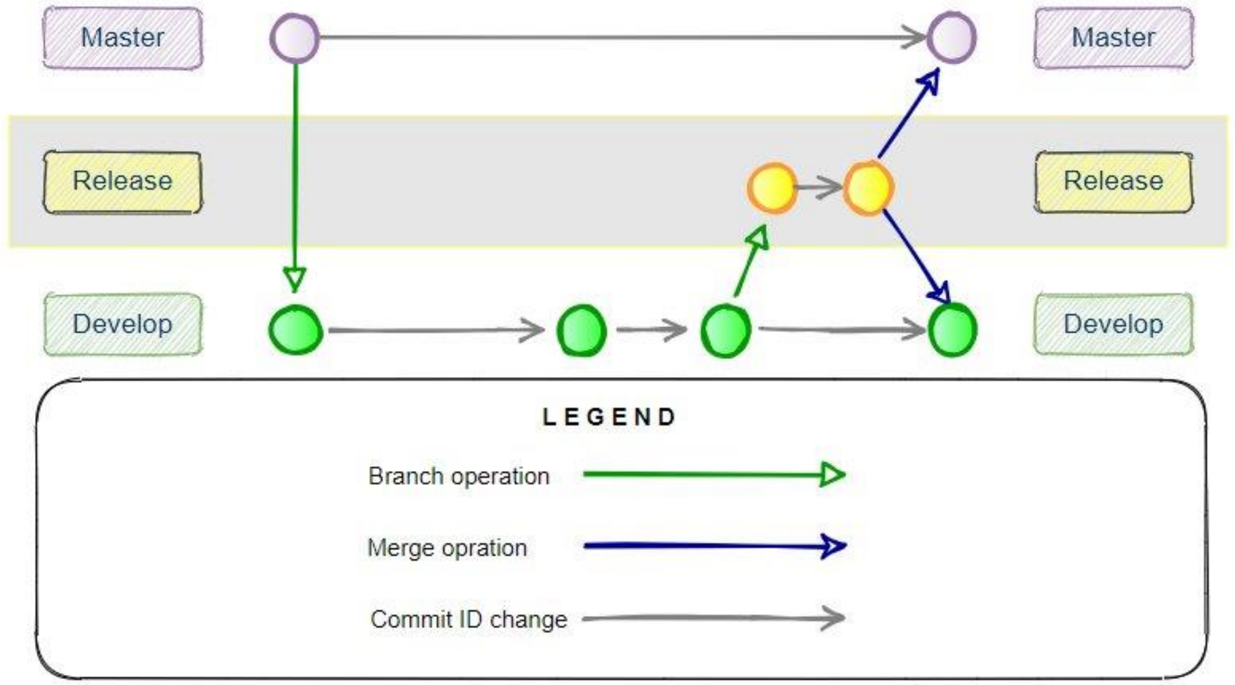
Core principle: A dedicated branch is created for each release, allowing for bug fixes and minor changes without affecting the main development branch.

Benefits: Better control over releases, reduced risk of introducing bugs into the main branch.

Challenges: Increased complexity, potential for merge conflicts.

TechPro

Release Branching Strategy



TechPro

Feature Branching

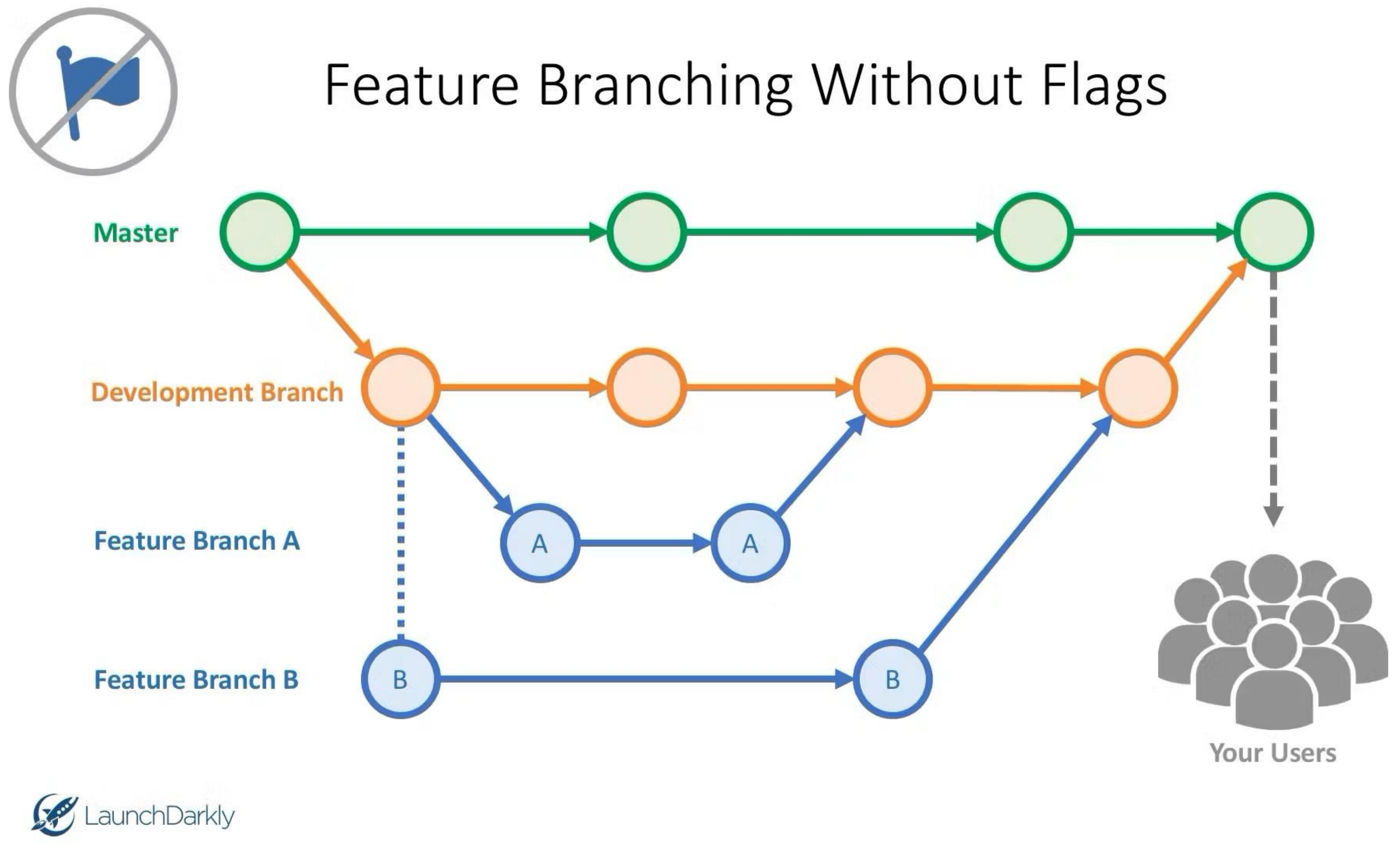
Core principle: Developers create separate branches for each new feature or bug fix.

Benefits: Isolation of changes, easier code review, and reduced risk of breaking the main branch.

Challenges: Increased overhead of managing multiple branches, potential for merge conflicts.

TechPro

Feature Branching Strategy



TechPro

Tasks and hotfixes Branching

- **Task branching :** also known as issue branching, directly connects those issues with the source code.

- **Hotfix branching:** A temporary branch created to address critical bugs in a released version.

TechPro

GitHub User Interface Demo

https://github.com/dashboard

GitHub Documentation Review

https://docs.github.com/en

TechPro

Working with GitHub

- Configure Git
- Create GitHub account
- Connect to GitHub
- Download code
- Clone code
- Fork
- Open Codespaces

TechPro

By the end of this Course

1. Understand the use of GitHub and be familiar with the UI
2. Create a GitHub account, clone and fork your first code in VS Code
3. Understand the most common Branching Strategies
4. You are familiar with developer’s collaboration with Git and GitHub
5. Work on a git repository with others and request your first pull

TechPro

Homework 1

https://www.w3schools.com/git/git_exercises.asp

Feedback Discussion

TechPro

Thank you

TechPro