

Working from Branches with GitHub Desktop



Stefan Roman

DEVOPS ENGINEER

medium.com/@stefanroman



What Is Git Branch?



Branches prevent interfering

Allow working on multiple features

Allow multiple people contributing

Switching between branches is lightweight

What Can We Do with Branches?



Publish branches

Commit into branches

Merge branches

Compare branches

Rename or delete branches

Rebase branches



What Is a Git Commit?



Adding identifier to the latest code change

Commits make up entire git repositories

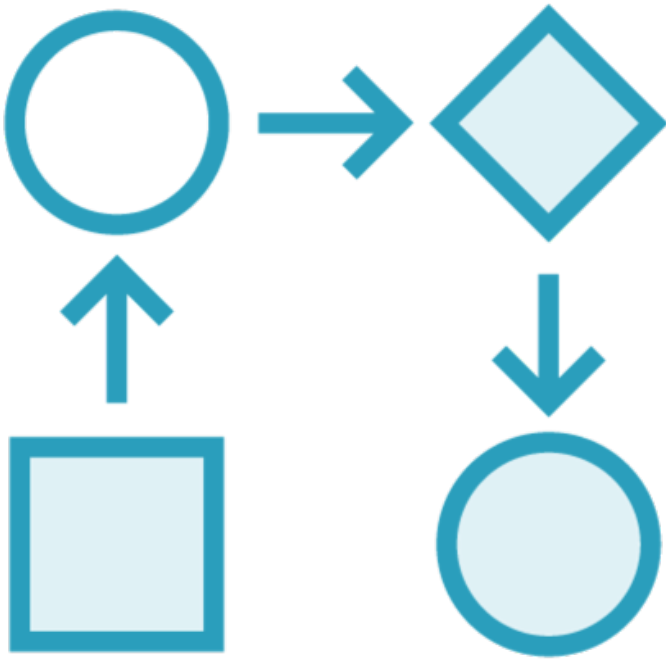
- Easy to rollback code

Local repository change only

- Push commits to publish

Identified with a hash identifier

Git Workflows



Centralized workflow

- One branch
- Prone to conflicts

Feature branch workflow

- Features utilize different branches
- Master branch is never broken
- Allows pull requests

Demo



Adding files into a repository attached in module resources

Selecting which lines to commit

Reverting commits

Commit history

Best practices for commit messages and commit descriptions



Demo



Working in team on repository with a single branch

Learning about repository origin

Simulating conflict and repair afterwards

Git actions such as pushing, pulling and fetching



Demo



Using multiple branches on our
picture_downloader repo

Creating branches and comparing them

Publish branches and have a look on
GitHub Cloud



Summary



What branches are and benefits of using them

Default branches for repositories

What commits are and how they work

Centralized and feature branch workflows



Summary



How to commit changes via GitHub Desktop

- Commit messages
- Commit descriptions

Revert unwanted commits

Utilize Centralized Workflow

Producing and fixing git conflict

Various git actions

Used Feature Branch Workflow

Created branches and compared them

