

# *Workshop Git & GitHub*

Xander Warszawski

**GitHub**



**IEEE Student Branch**  
**KU Leuven Campus Brugge**

# ***1. Introduction***

# Goals

---

- Understanding & being able to use **Git**
  - Basic commands using command line
  - **SourceTree** for 'daily' use
- Understanding & being able to apply some **Git branching strategies**
  - **GitFlow**
  - **GitHub Flow**
  - **Trunk-based development**
- Being able to use **GitHub**

# Who am I?

---

- Xander Warszawski
- Master student Engineering Technology ICT
- IEEE SB KU Leuven Campus Brugge Chair
- GitHub Campus Expert
  
- For any questions: [xander@xdoubleu.com](mailto:xander@xdoubleu.com)



# GitHub Campus Expert

**Campus Experts** are **student leaders** that **strive to build diverse and inclusive spaces** to **learn skills, share their experiences**, and **build projects** together. They can be **found across the globe** leading **in-person and online conferences, meetups**, and **hackathons**, and **maintaining open source projects**.

More info: <https://education.github.com/experts>

Are you a 2<sup>nd</sup> year student and do you want to become the next Campus Expert? Please contact me!

# *GitHub Student Developer Pack*

- [https://education.github.com/discount\\_requests/student\\_application](https://education.github.com/discount_requests/student_application)



## ***2. Install all tools***

# *Git install*

---

- <https://git-scm.com/download>
- Default everything



# *Sourcetree install*

---

- <https://www.sourcetreeapp.com/>
- Skip registration
- Install git
- SSH Key: No

# 3. *Git*

# Git?

---

- **Git** is a **VCS** (**V**ersion **C**ontrol **S**ystem)
  - Allows developers to store all versions of their code as one (= repository)
  - *Branching and merging*

# Terminology

- **Branch**: parallel version in repo
- **Commit**: saving changes in branch
- **Merge**: merge branches or repo
- **Pull Request** (GH) / Merge Request (GL): request to merge branch or fork
- **Repo**(sitory): project 'folder' (code, docs); has all versions
- **Clone**: copy of repo, mostly offline
- **Fetch**: getting last version of online repo
- **Pull**: 'download' latest commits from remote repo to clone
- **Push**: 'upload' committed changes to remote repo
- **Reverse**: undo changes made in commit(s), useful for when application breaks in production
- **Reset**: remove commit, using this is a bad practice
- **Cherry pick**: pick a certain commit out of the change history
- **Fork**: start new project from existing

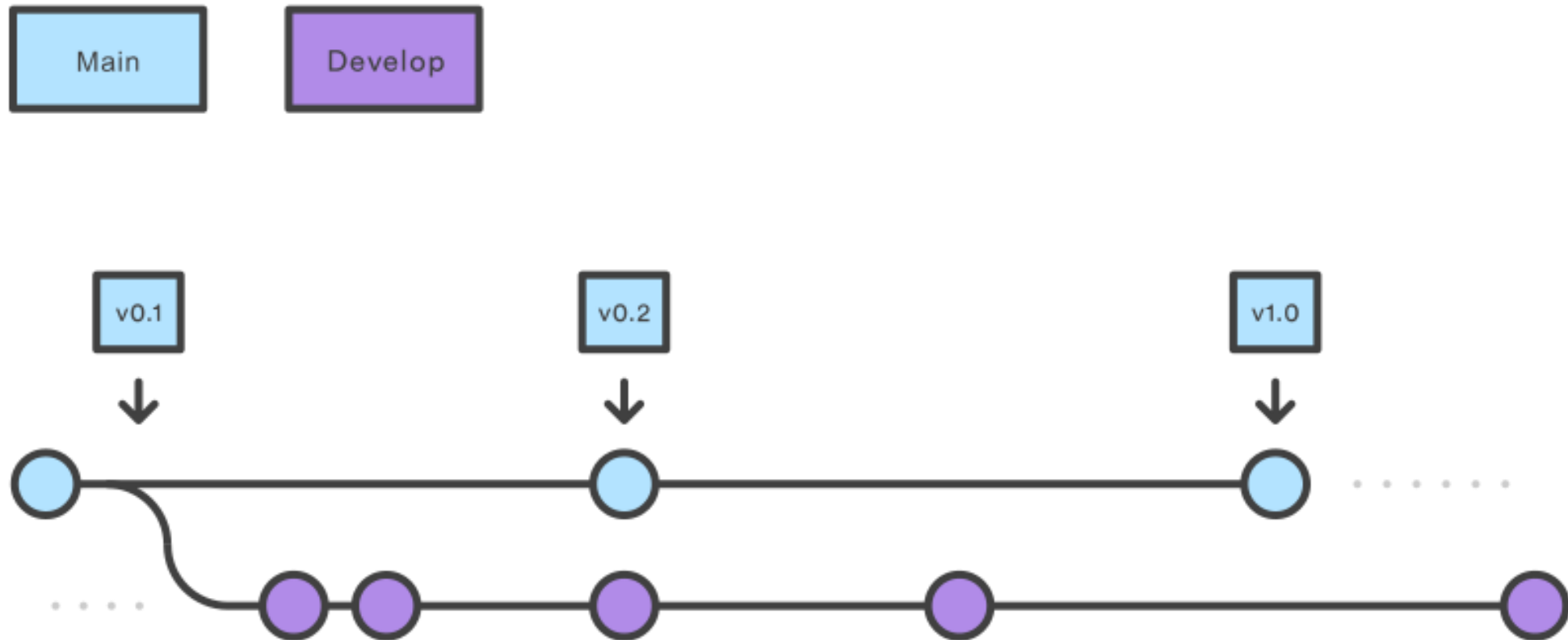
# Questions?

---

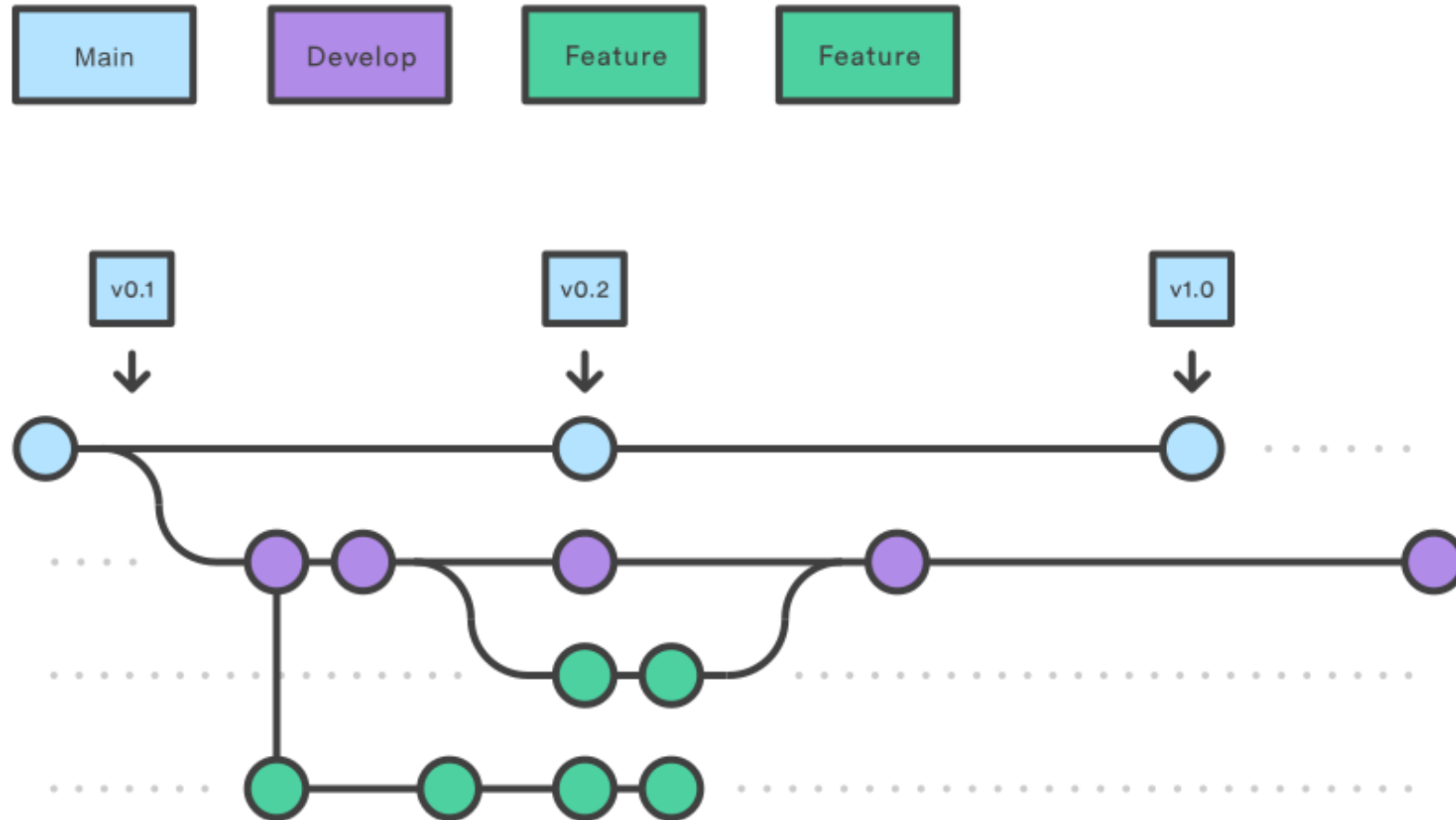
- If you want to practice Git commands:
  - <https://learngitbranching.js.org>

# **4. *Git branching strategies***

# GitFlow: main & dev

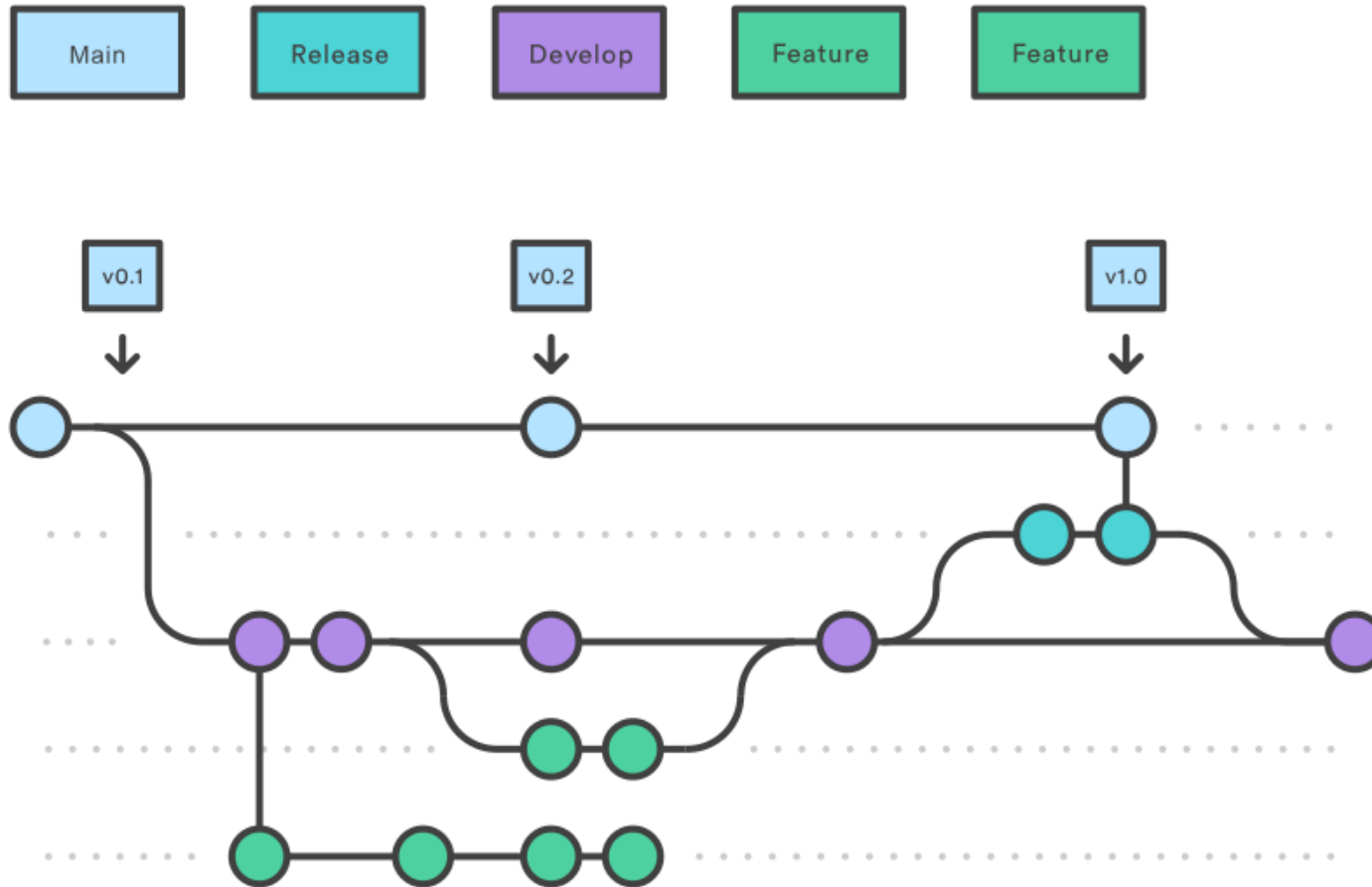


# GitFlow: feature

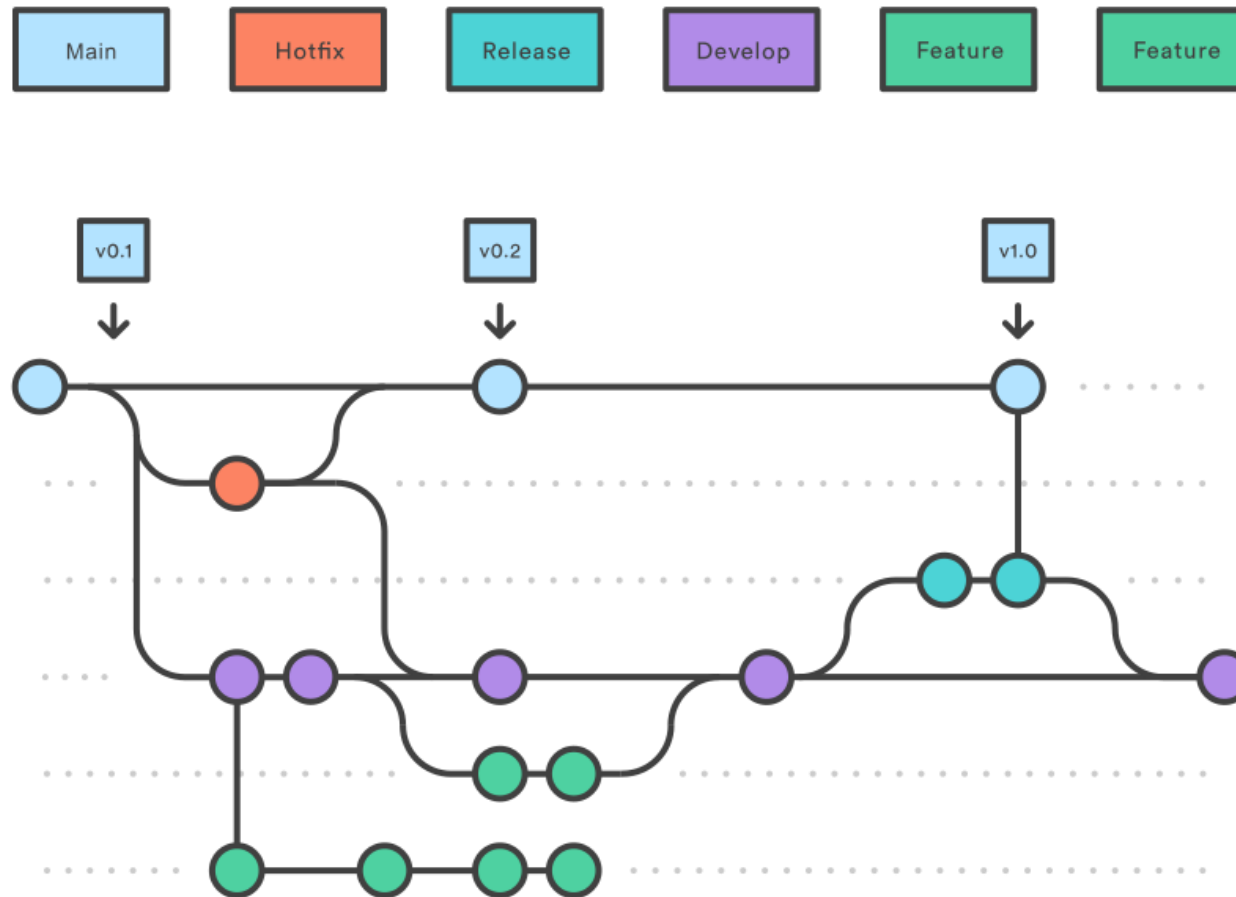




# GitFlow: release



# GitFlow: hotfix

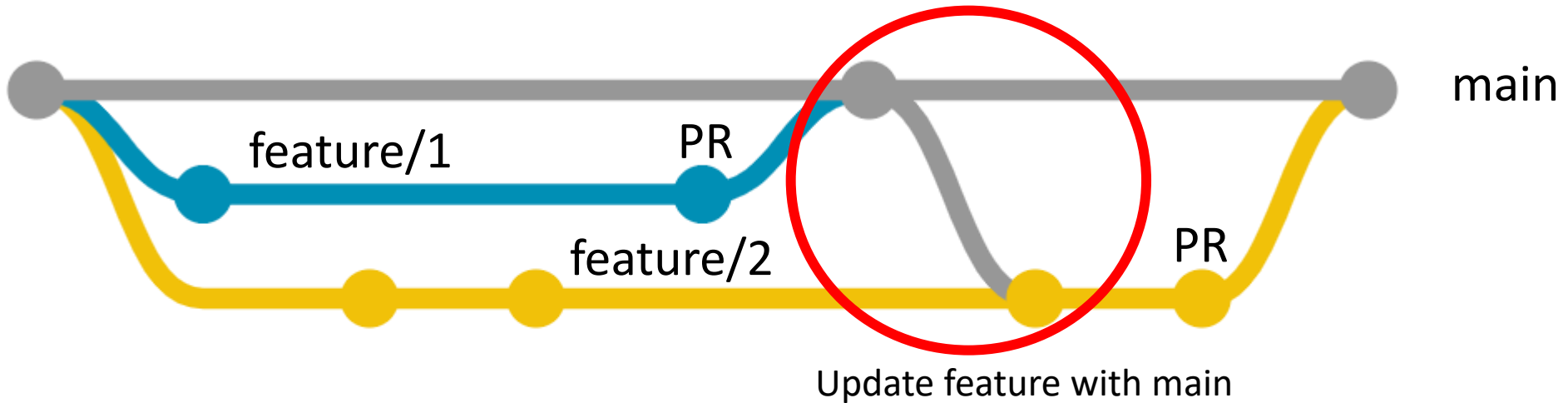


# *Pros & cons of GitFlow*

---

- Pros:
  - Lots of control on which code gets in the codebase
- Cons:
  - Slow & complex flow

# GitHub Flow/feature branching

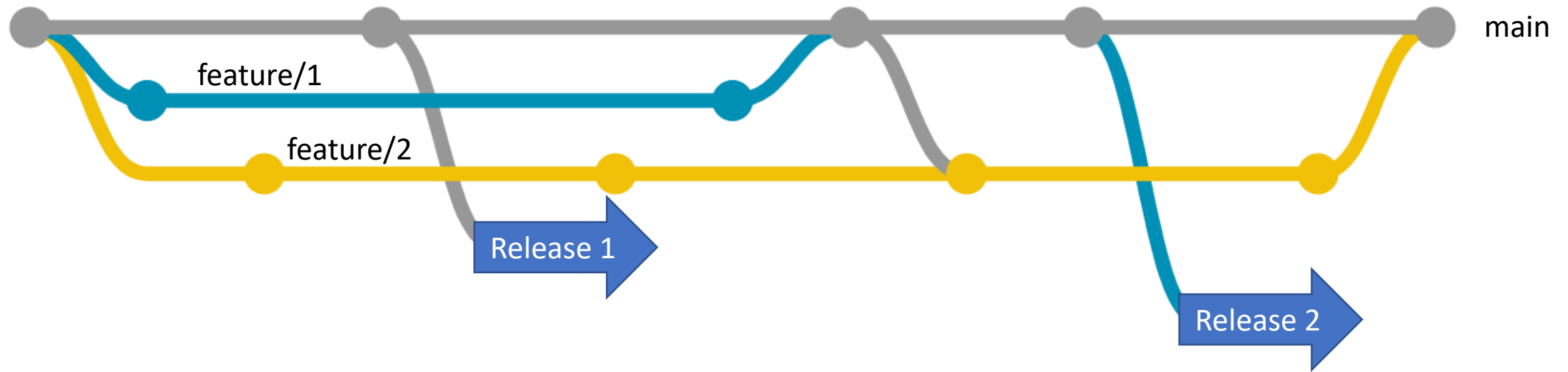


# *Pros & cons of GitHub Flow*

---

- Pros:
  - Fast & streamlined
  - Continuous delivery
- Cons:
  - Pull requests with reviews can slow down flow

# Trunk-based development



# *Pros & cons of trunk-based development*

---

- Pros:
  - Very simple
  - Scales
- Cons:
  - No code reviews before merge

# *Recommended flow?*

---

- GitHub flow with code reviews and build automation on pull request



# **5. *GitHub***

# GitHub?

---

- **GitHub** is a **Hub**
  - Hub? **A hub** airport is an airport used by one or more airlines **to concentrate** passenger **traffic and** flight **operations**.
  - So? GitHub is a **collection of Git repositories**
    - Also publishing & collaboration tool

# **6. *In Action***

# Git CLI

- git clone [url] & git init
- git add [file]/.
- git reset [file]/.
- git status
- git commit -m "[message]"
- git push & git pull
- git branch
- git checkout -b [branch-name]
  - Or: git branch [branch-name] + git checkout [branch-name]
- git merge [branch] (merges [branch] into current branch)

# GitHub (1)

---

- Create new repo
- Fork repo
- .gitignore: <https://toptal.com/developers/gitignore>
- GitHub actions

# GitHub (2)

---

- Interesting settings:
  - General:
    - Always suggest updating PR branches
    - Auto delete head branches
  - Branches:
    - Branch protection rules:
      - Require PR
      - Require Status Checks to pass
        - Require branch to be up-to-date
      - Require linear history
  - Code security & analysis
    - Enable all

# SourceTree

---

- Clone
- Pull
- New branch
- Commit
- Merge
- Push
- PR
- Fetch
- Rebase
- Cherry pick
- Reverse & Reset
- Stash
- Terminal + explorer