

# Team work distribution

Project management

# Project management



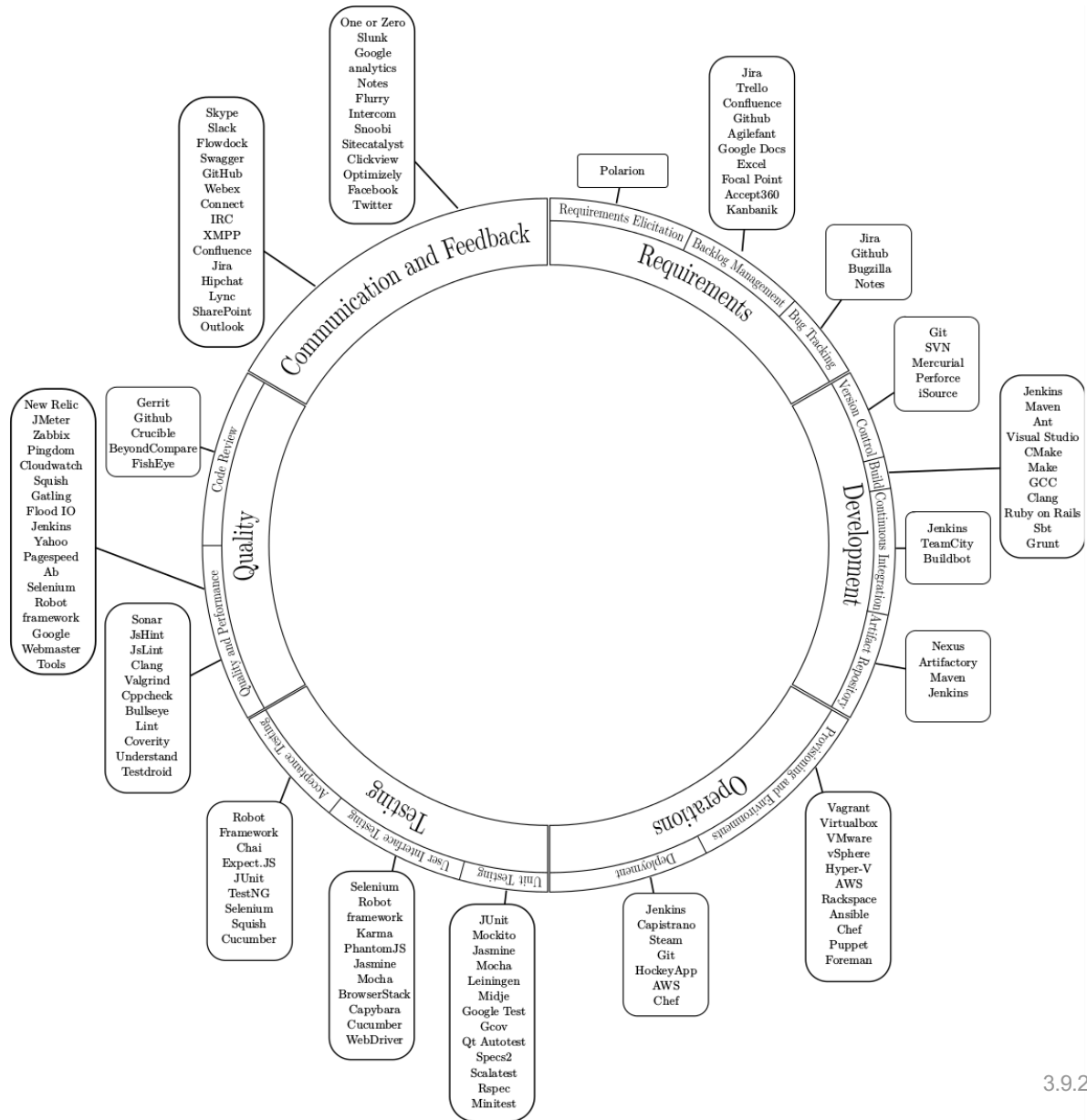
Kuva: James Shore, 2006

# Coder's normal day

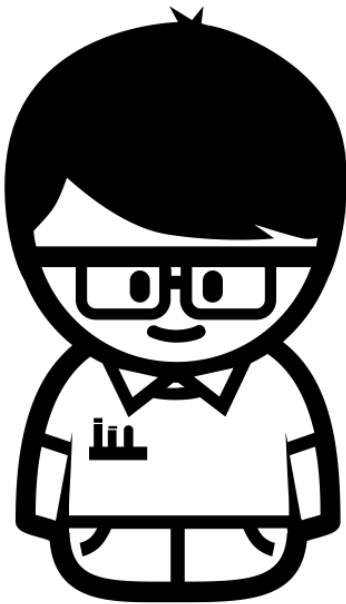


Fig: Kristina Alexanderson(CC BY-NC-ND 2.0)

# Programming tools



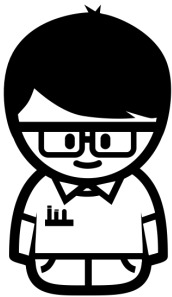
# File management (dev)



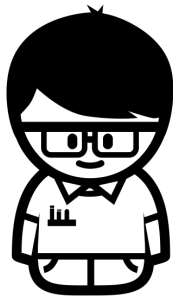
Created by Javier Cabezas  
from Noun Project

- Command-line interpreter
  - Files
  - Directories
  - Commands and scripts
- IDE
  - Editor
  - Compiler

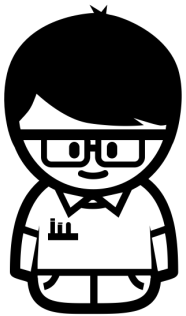
# File management (team)



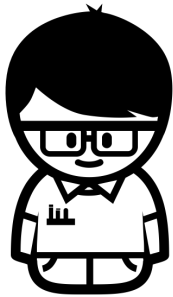
Created by Javier Cabezas  
from Noun Project



Created by Javier Cabezas  
from Noun Project



Created by Javier Cabezas  
from Noun Project



Created by Javier Cabezas  
from Noun Project

- Version control
  - Who?
  - What?
  - Permission to remove!
- Version control includes
  - Code
  - Scripts
  - Documentation
  - ...

# Using version control

- Local, centralized, or distributed
- Tool for communication

```
git commit -a -m "These aren't the droids..."
```

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

Kuva: XKCD (CC BY-NC 2.5)

AS A PROJECT DRAGS ON, MY GIT COMMIT  
MESSAGES GET LESS AND LESS INFORMATIVE.

# Git and GitLab

```
[otekn@arokorppi] ~/kurssipriva > git pull origin
remote: Counting objects: 37, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 29 (delta 15), reused 0 (delta 0)
Unpacking objects: 100% (29/29), done.
From gitlab.rd.tut.fi:otekn_2015-2016/kurssipriva
 4d83fe9..963b65b  master    -> origin/master
Updating 4d83fe9..963b65b
Fast-forward
 harkkatyo/Nysse0su/KurssinPuoli/KurssinPuoli.pro | 20 +-
 harkkatyo/Nysse0su/KurssinPuoli/offlinedata.qrc | 6 +
 .../Nysse0su/KurssinPuoli/offlinedata/bus.json | 488 ++++
 .../Nysse0su/KurssinPuoli/offlinedata/station.json | 2428 ++++++
 harkkatyo/Tyooohje/Tyooohje.htm | 272 +++
 harkkatyo/Tyooohje/Tyooohje_files/lemonade.css | 111 +
 harkkatyo/Tyooohje/Tyooohje_files/respond.js | 6 +
 harkkatyo/Tyooohje/Tyooohje_files/styles.css | 117 +
 harkkatyo/notes.txt | 4 +
 paivakirja.txt | 15 +-
 public_html/luennot.shtml | 9 +-
11 files changed, 3465 insertions(+), 11 deletions(-)
create mode 100644 harkkatyo/Nysse0su/KurssinPuoli/offlinedata.qrc
create mode 100644 harkkatyo/Nysse0su/KurssinPuoli/offlinedata/bus.json
create mode 100644 harkkatyo/Nysse0su/KurssinPuoli/offlinedata/station.json
create mode 100644 harkkatyo/Tyooohje/Tyooohje.htm
create mode 100644 harkkatyo/Tyooohje/Tyooohje_files/lemonade.css
create mode 100644 harkkatyo/Tyooohje/Tyooohje_files/respond.js
create mode 100644 harkkatyo/Tyooohje/Tyooohje_files/styles.css
create mode 100644 harkkatyo/notes.txt
```



# Version control system: Git

- Distributed
- Command line is your friend
- Basic actions
  - `git clone git@course-gitlab.tuni.fi:viikkoharkat.git`
  - `git pull origin master`
  - `git add droids.txt`
  - `git commit -m "not looking for"`
  - `git push origin master`

# Version control system: Git

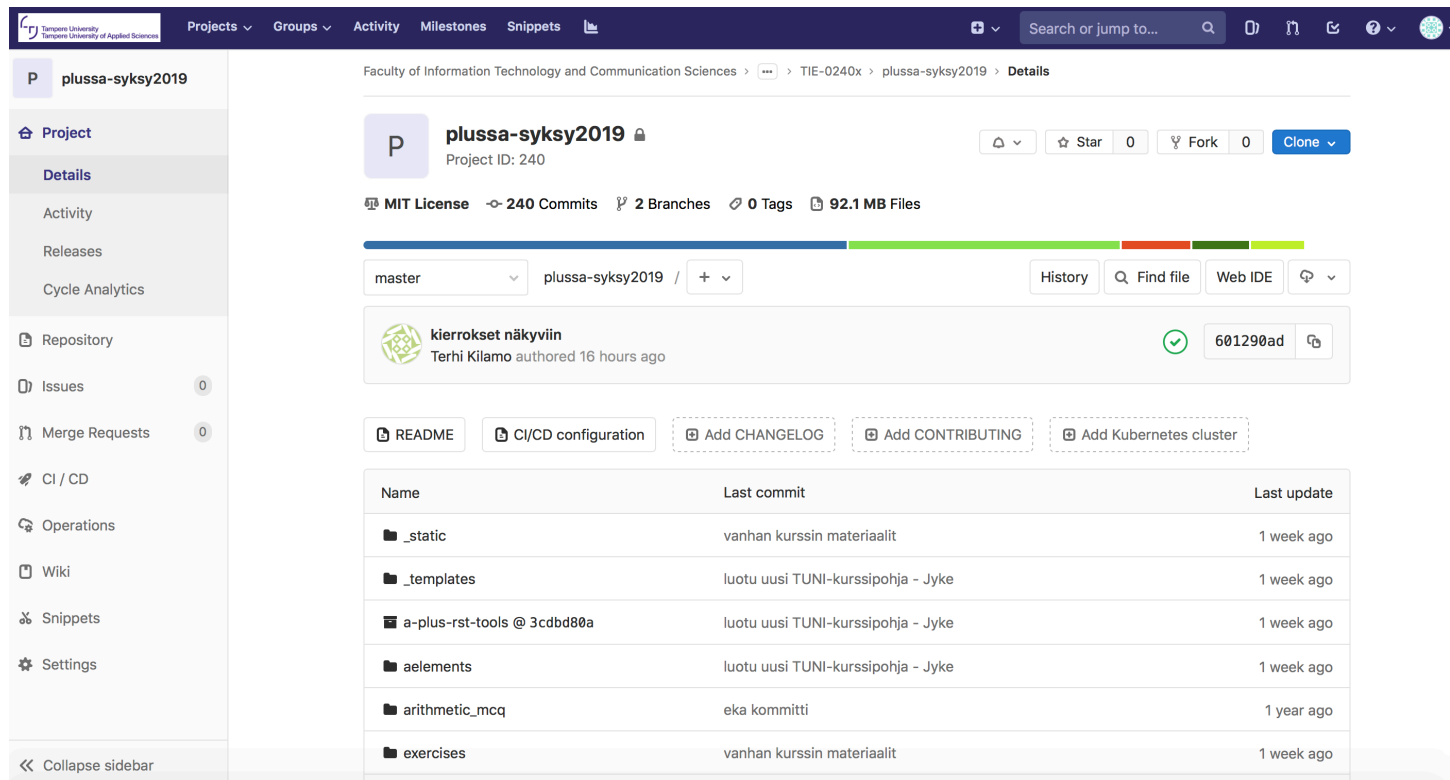
- Basic actions

- `git clone git@course-gitlab.tuni.fi:repo.git`
- `git pull origin master`
- `git add droids.txt`
- `git commit -m "not looking for"`
- `git push origin master`

# Git commands

- git config
- git status
- git log
- git tag
- git help
- git stash
- git merge
- git checkout
- git branch
- git blame

# GitLab: course-gitlab.tuni.fi



The screenshot shows the GitLab interface for the project 'plussa-syksy2019'. The left sidebar contains navigation links: Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Wiki, Snippets, and Settings. The main content area displays the project details, including the MIT License, 240 Commits, 2 Branches, 0 Tags, and 92.1 MB Files. A progress bar is visible below the statistics. The 'master' branch is selected, and the project path is 'plussa-syksy2019 / +'. A commit by Terhi Kilamo is highlighted with a green checkmark and the commit hash 601290ad. Below the commit list, there are buttons for README, CI/CD configuration, Add CHANGELOG, Add CONTRIBUTING, and Add Kubernetes cluster. A table lists the project's files and their last commit details.

Name	Last commit	Last update
_static	vanhan kurssin materiaalit	1 week ago
_templates	luotu uusi TUNI-kurssipohja - Jyke	1 week ago
a-plus-rst-tools @ 3cddb80a	luotu uusi TUNI-kurssipohja - Jyke	1 week ago
aelements	luotu uusi TUNI-kurssipohja - Jyke	1 week ago
arithmetic_mcq	eka kommitti	1 year ago
exercises	vanhan kurssin materiaalit	1 week ago

# The freedom of version control

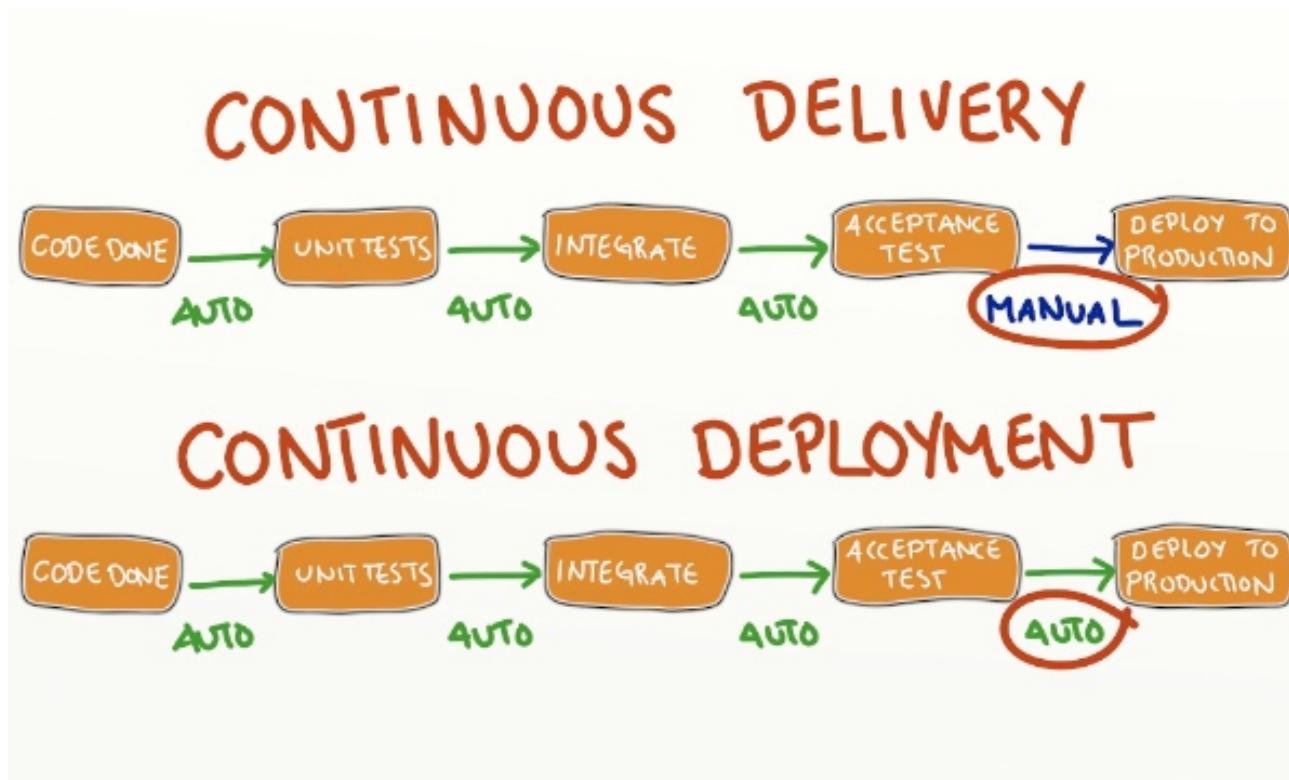


Kuva: XKCD (CC BY-NC 2.5)

# Team work distribution

Integration and conflicts

# Continuous development



Kuva: Yassal Sundman, 2013





# Conflicts in version control



Kuva: Columbia.edu, Zach Newman

# Conflicts

Editing the same file at the same time, and trying to merge the changes → poor Git does not know what to do and asks for help

Typical conflicts:

- Editing the same content
- Editing a removed file

# Conflicts

How to know about a conflict?

```
➤ less solarSystem.txt
  The number of planets is
  <<<<<<< HEAD
  nine
  =====
  eight
  >>>>>>> branch-a
```

# Resolving conflicts

- Resolve the conflict by yourself
- Ask the other party of the conflict to resolve it

```
$ git status
# On branch branch-b
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add ..." to mark resolution)
#
# both modified:      styleguide.md
#
no changes added to commit (use "git add" and/or "git commit -a")
```

# Removed files

- Keep:

```
git add planets.txt  
git commit -m "Resolved merge"  
or
```

- Remove:

```
git rm planets.txt  
git commit -m "Resolved merge"
```

# Competitive changes

- Decide which modification is valid: your one, the other one or a combination of these two
- Remove conflict-resolution markers  
<<<<<<, =====, >>>>>> and modify as intended  
git add .  
git commit -m "Resolved merge"

# Branches in version control

- Remote repository
- Coding in master branch (default name)
- Branch by team
- Branch by feature
- Branch for release
- ...

# **git remote add <shortname> <url>**

➤git remote

origin

➤git remote add up git@gitlab.rd.tut.fi:kurssi.git

➤git remote -v

up git@gitlab.rd.tut.fi:kurssi.git (fetch)

up git@gitlab.rd.tut.fi:kurssi.git (push)

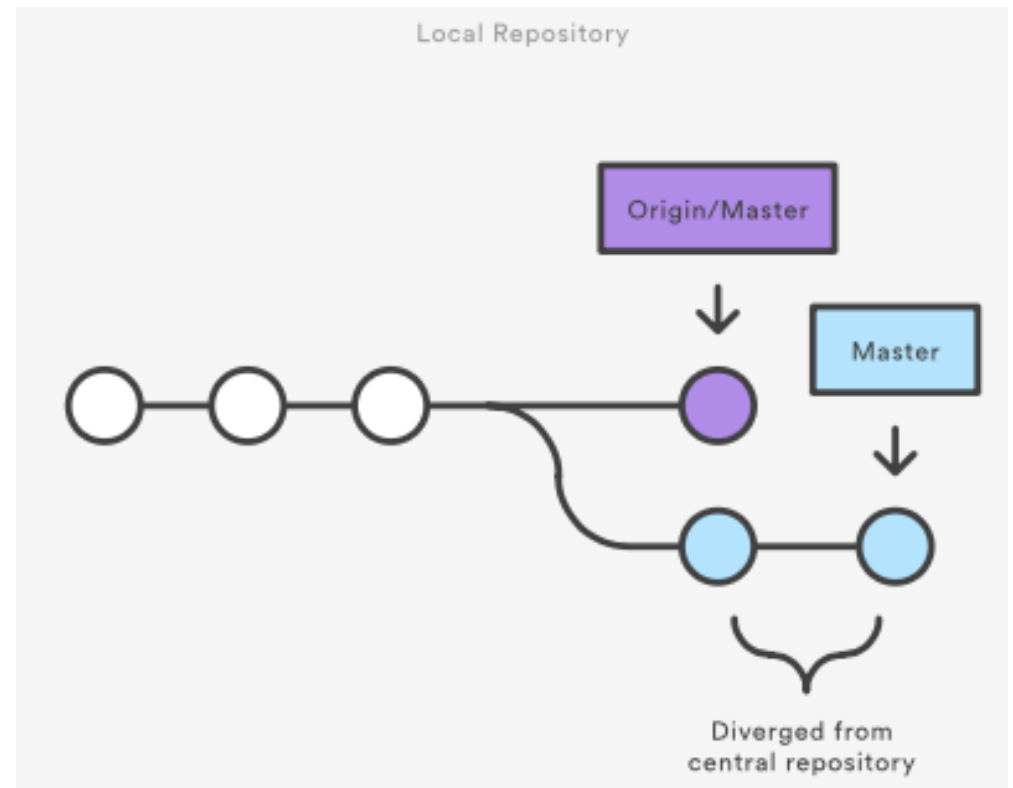
origin git@gitlab.rd.tut.fi:r1234.git (fetch)

origin git@gitlab.rd.tut.fi:r1234.git (push)



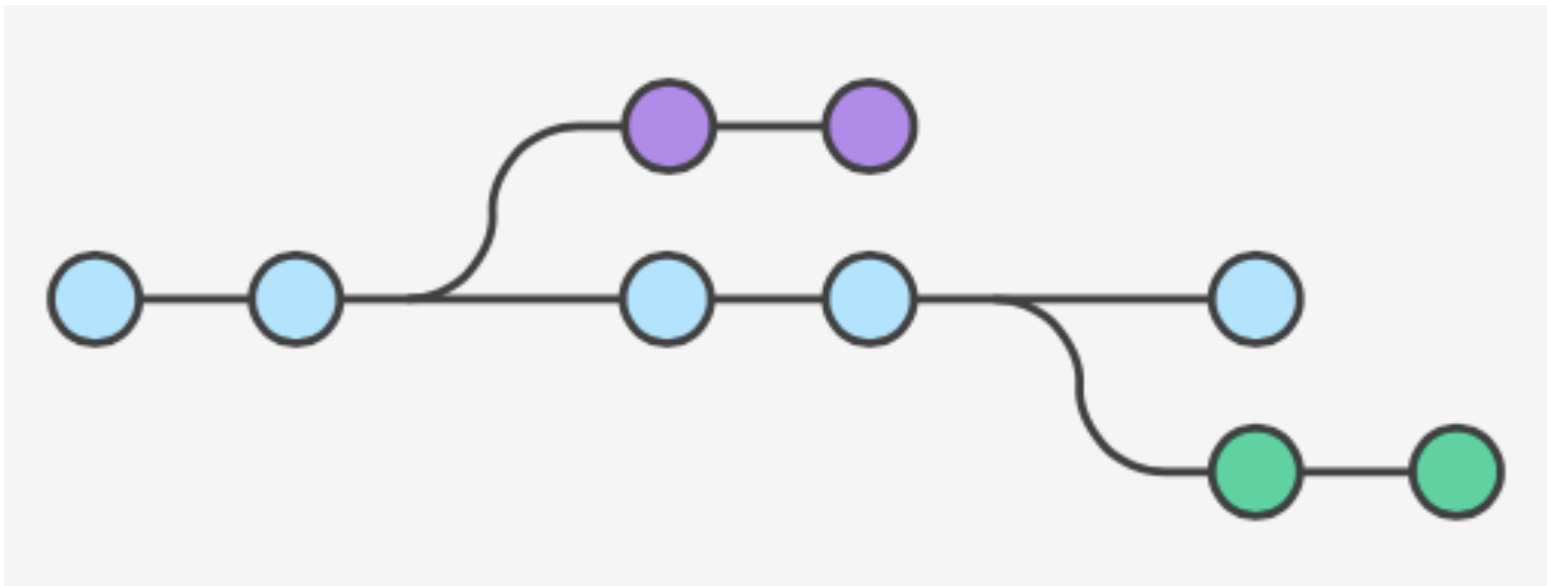
# Coding in master branch

- Commit at least once a day
- Plan and implement modifications in small steps
- History is sacred and immutable



# Branch by Feature

- Each feature has a branch

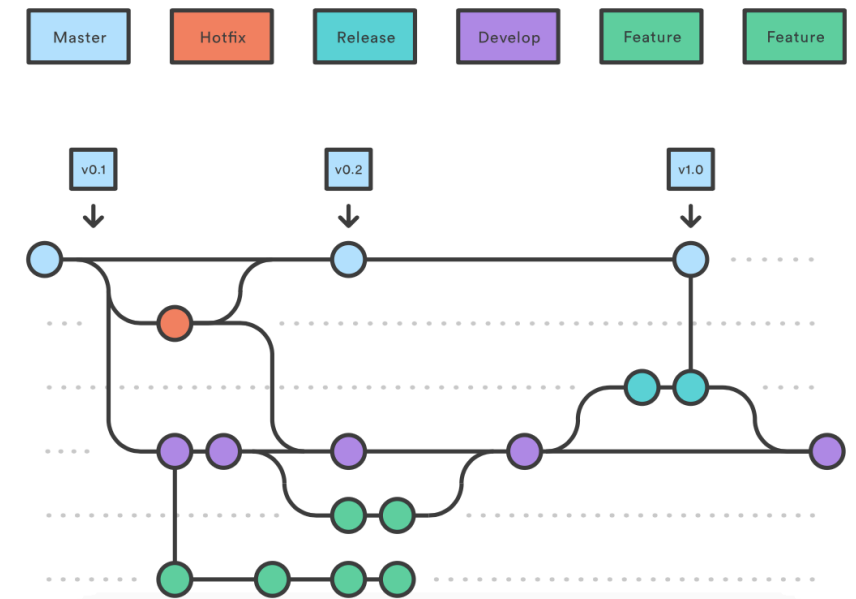


# Branch by Team

- Each team has a branch
- When the branch of a team is stable, merge it to the main branch

# Branch for Release

- Own branch for the release version, development continues in master branch



# Branches on Programming 3

On this course:

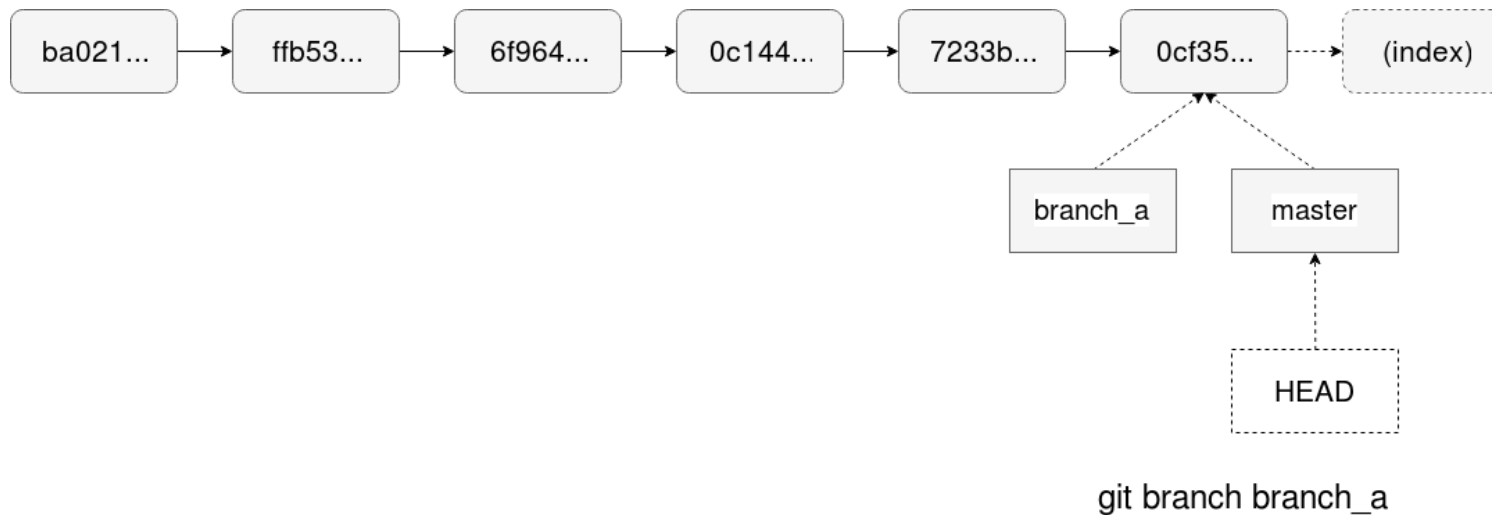
- You need a remote repository
- You do NOT need branches
- Branches may be useful in certain situations

# Branches on Programming 3

1. to keep extra parts apart from product code
  - Branches justifiable if you implement extra features in the project work
2. to give you time to work on your code with maintaining good version history
  - Master will stay compilable
  - All your work will stay under version control

# Branches in Git

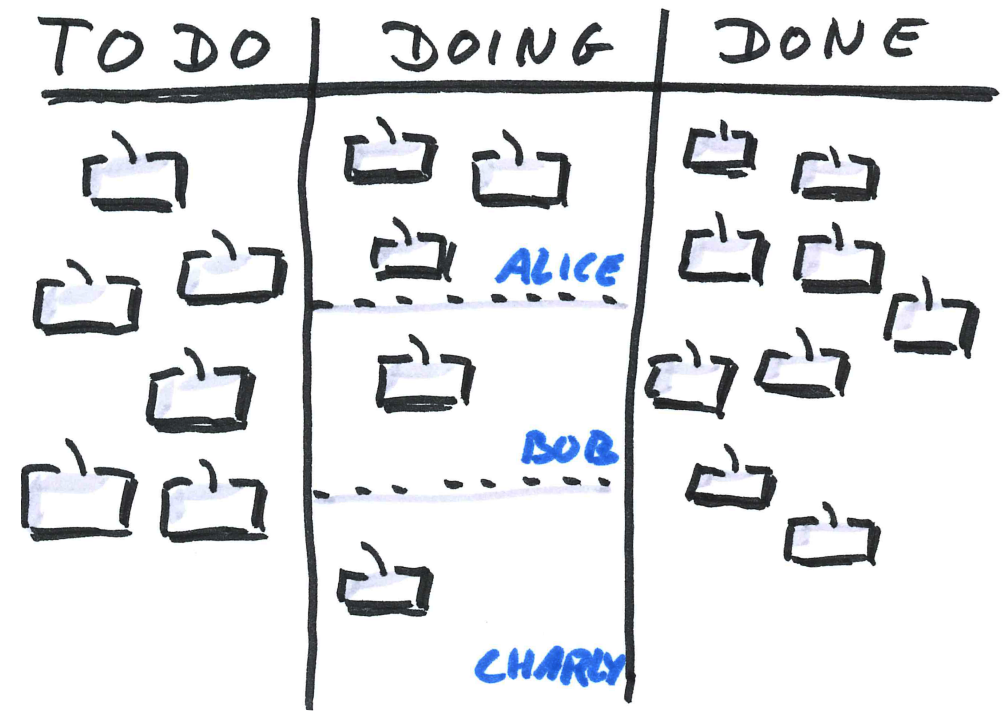
- List all branches: `git branch`
- Create branch: `git branch name`
- Move to branch: `git checkout name` or
- Create and move to branch: `git checkout -b name`



# Work distribution

Simple methods are the best

- Kanban board
- trello.com





# Finally

