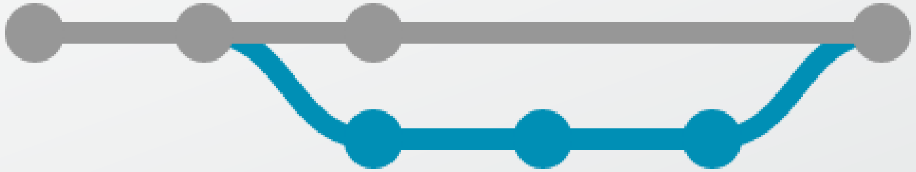


GIT WORKSHOP

May all the versions be with you

18.04.2018
Nikita Basargin

ROBOY.ORG

COMMON CHALLENGES

Scenario: Many developers work on the same code.

How can I …

… do changes without breaking something else?

… share my code with others efficiently?

Teamwork

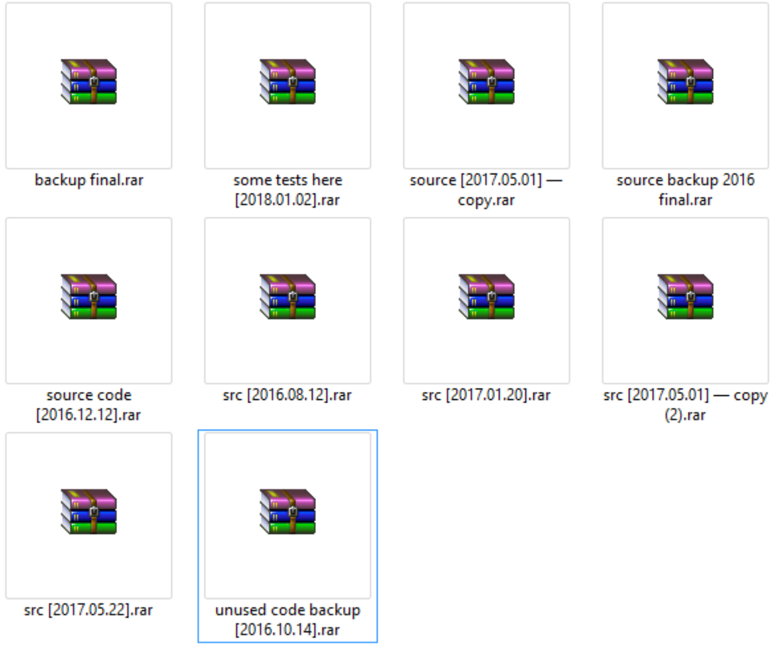
… make backups requiring little space?

… restore a previous version if something breaks?

Backups

NAIVE SOLUTIONS

Dropbox



backup final.rar

some tests here [2018.01.02].rar

source [2017.05.01] — copy.rar

source backup 2016 final.rar

source code [2016.12.12].rar

src [2016.08.12].rar

src [2017.01.20].rar

src [2017.05.01] — copy (2).rar

src [2017.05.22].rar

unused code backup [2016.10.14].rar

BETTER SOLUTION

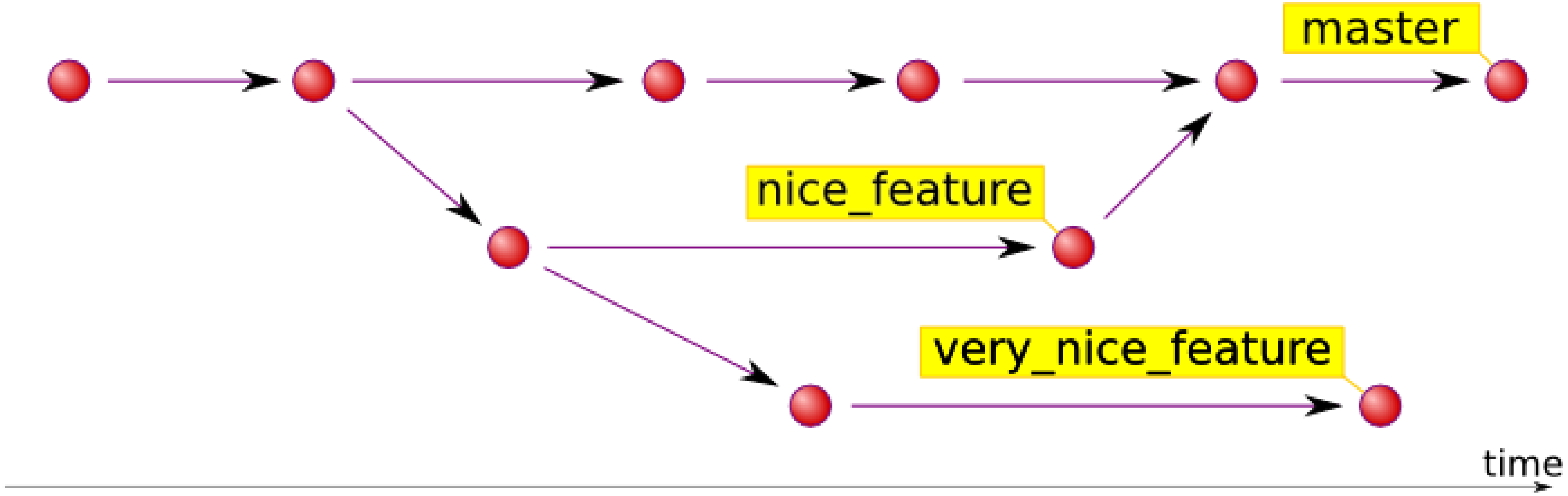
BUT…

“Whoah, I’ve just read this quick tutorial about git and oh my god it is cool.
I feel now super comfortable using it, and I’m not afraid at all to break something.”

– No one ever

IDEA

- Incremental versioning system
- Save your work in **commits**
- Develop new features on different **branches**



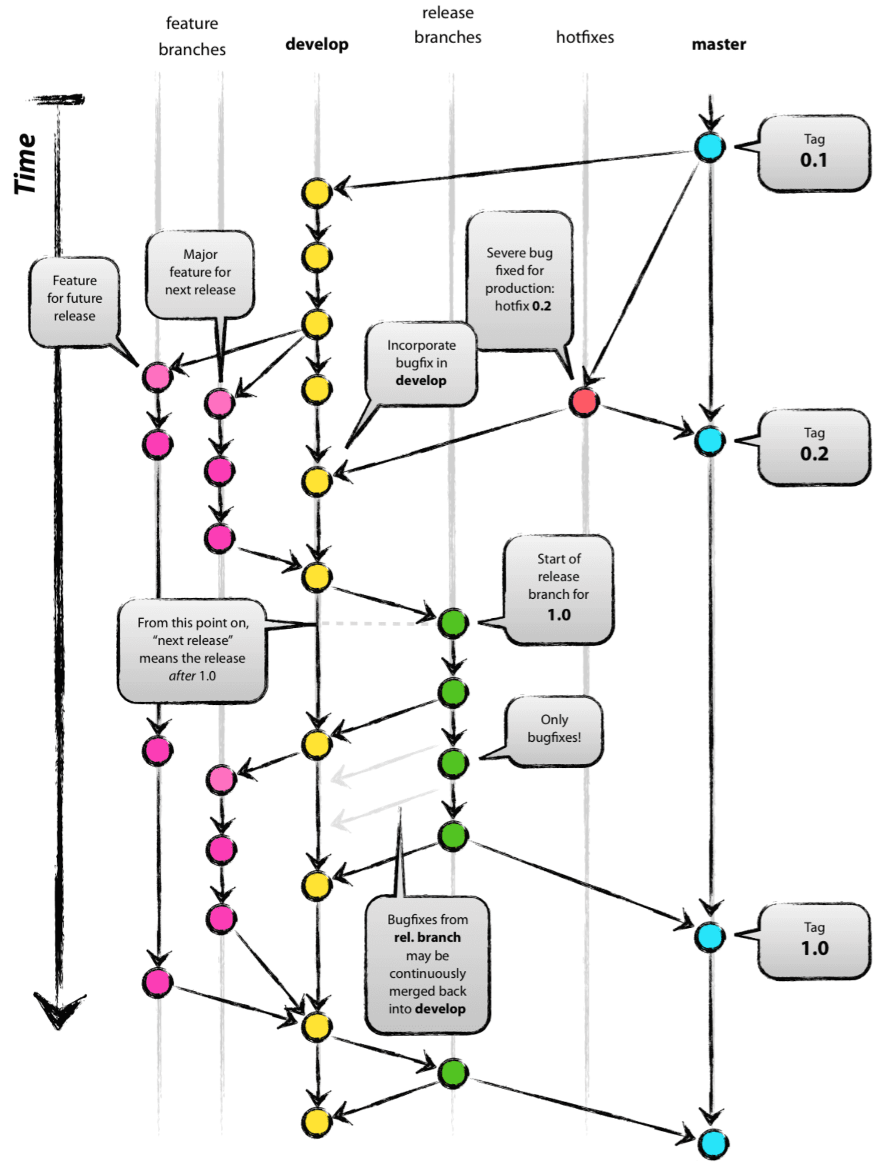
BRANCHING MODEL

Common branch conventions

- master: always stable code
- develop: current development
- feature: new (unstable) code

Read more here:

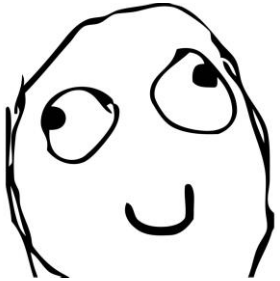
http://nvie.com/posts/a-successful-git-branching-model/


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

SOFTWARE

GUI user

- Easier to start
- Good overview

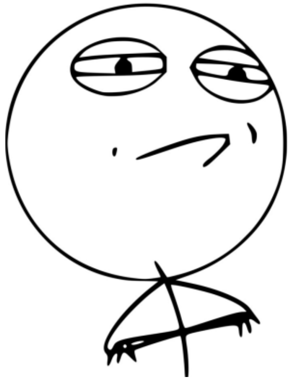


Tools:

- Sourcetree
- GitHub Desktop
- GitKraken

Command line user

- Better understanding
- More control



Tools:

- Command line

GIT COMMANDS

```
$ git init [project-name]
```

Creates a new local repository with the specified name

```
$ git clone [url]
```

Downloads a project and its entire version history

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

```
$ git status
```

Lists all new or modified files to be committed

```
$ git add [file]
```

Snapshots the file in preparation for versioning

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

```
$ git reset [file]
```

Unstages the file, but preserve its contents

`$ git commit -m "[descriptive message]"`

Records file snapshots permanently in version history

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

```
$ git branch
```

Lists all local branches in the current repository

```
$ git branch [branch-name]
```

Creates a new branch

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

```
$ git checkout [branch-name]
```

Switches to the branch and updates the working directory

```
$ git merge [branch]
```

Combines the branch’s history into the current branch

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

```
$ git log
```

Lists version history for the current branch

```
$ git fetch
```

Downloads all branches from the repository

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

GIT COMMANDS

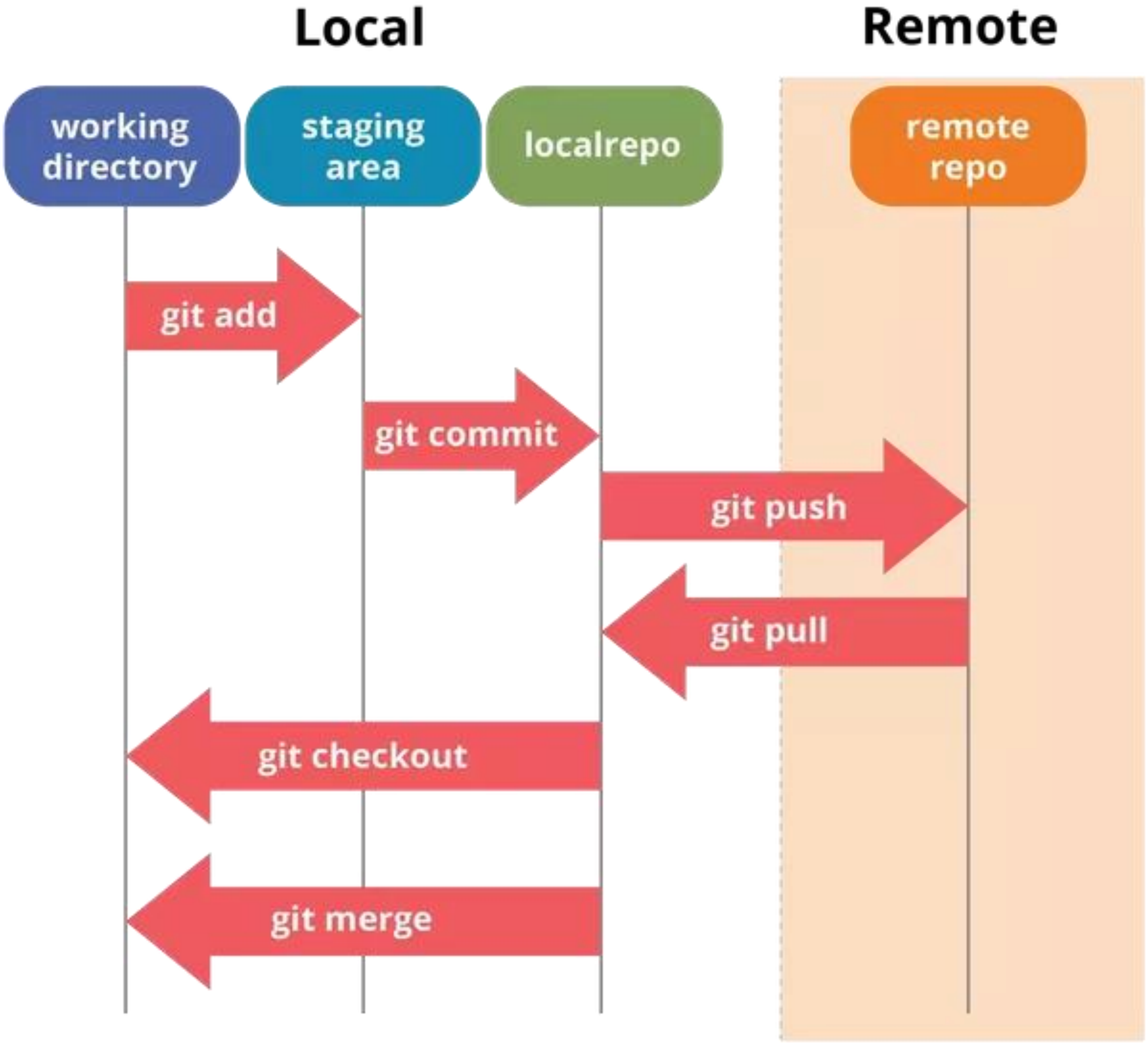
```
$ git push
```

Uploads all local branch commits to the git server

```
$ git pull
```

Downloads history and changes from the server

Cheat Sheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge

REMOTE SERVERS

GitHub

- Famous in the open source community
- Actively used in Roboy
  → https://github.com/Roboy

Alternatives for private/university projects

- Bitbucket
- GitLab

WORKSHOPS & TUTORIALS

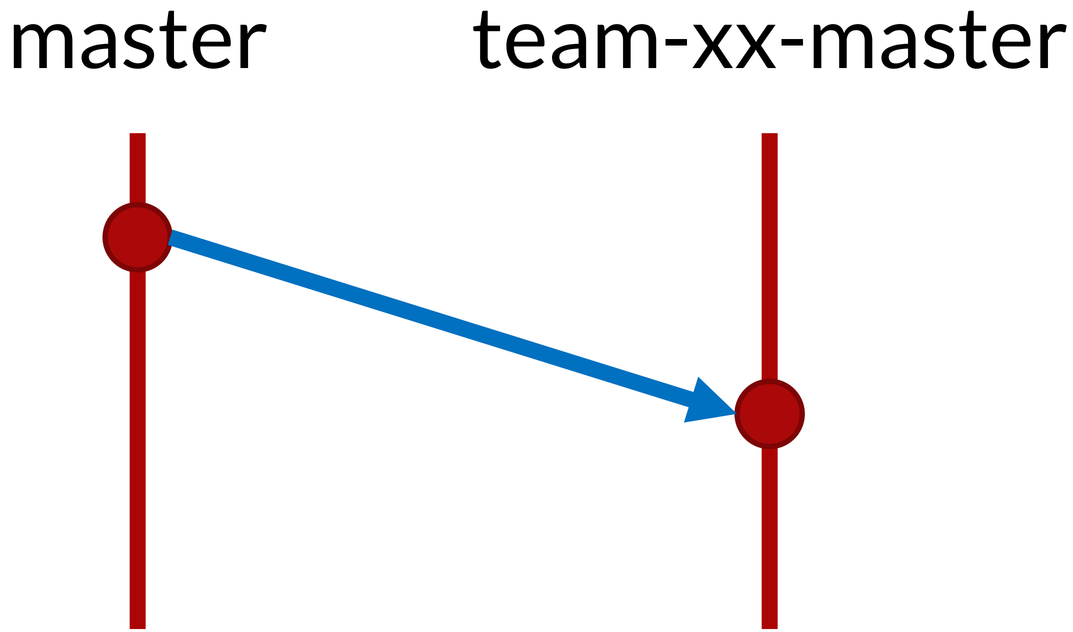
Roboy SS18 git workshop (for teams of three):

https://github.com/Roboy/roboy_git_tutorial_ss18

Interactive tutorial:

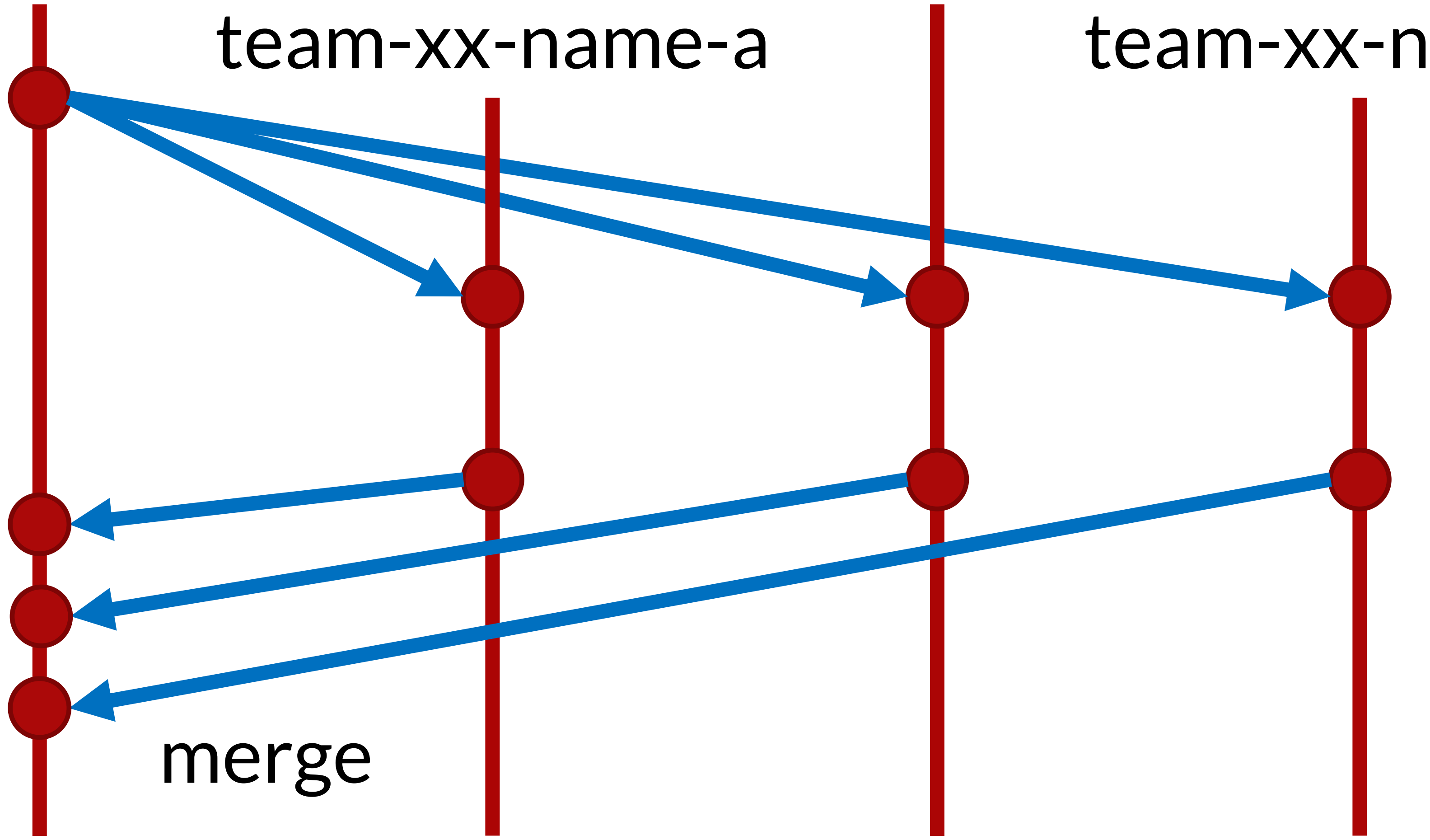
https://try.github.io/

ROBOY SS18 WORKSHOP – STEP 1



Create a new master branch for your team

ROBOY SS18 WORKSHOP – STEP 2



team-xx-master

team-xx-name-a

team-xx-name-b

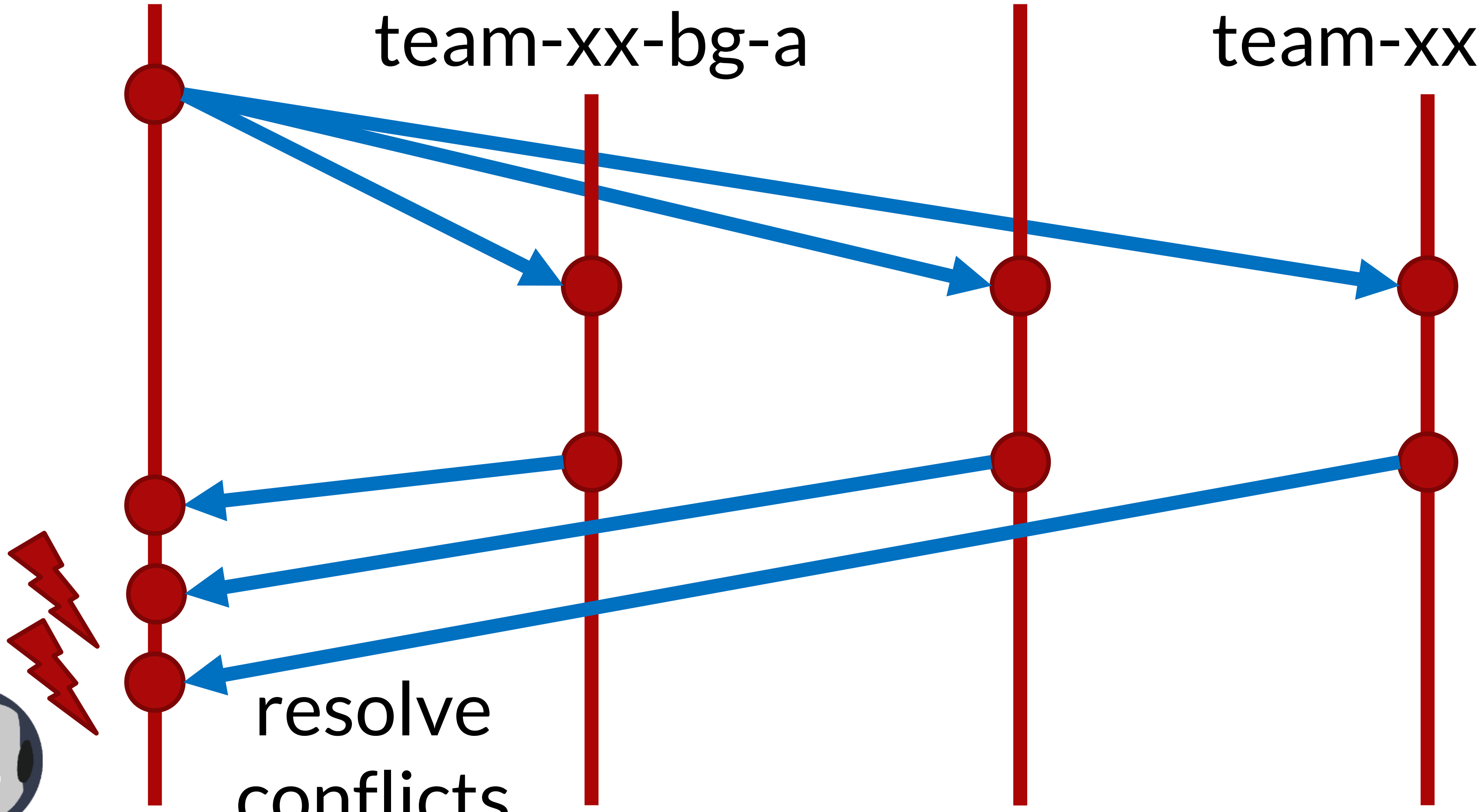
team-xx-name-c

new branch for each member

independent changes

merge

ROBOY SS18 WORKSHOP – STEP 3



team-xx-master

team-xx-bg-a

team-xx-bg-b

team-xx-bg-c

new branch for each member

conflicting changes

resolve conflicts

TIME TO HACK

https://github.com/Roboy/roboy_git_tutorial_ss18

In case of fire

1. git commit

2. git push

3. leave building

THANK YOU FOR YOUR ATTENTION!