



**University of
Zurich** ^{UZH}



**URPP Evolution
in Action**

URPP tutorial

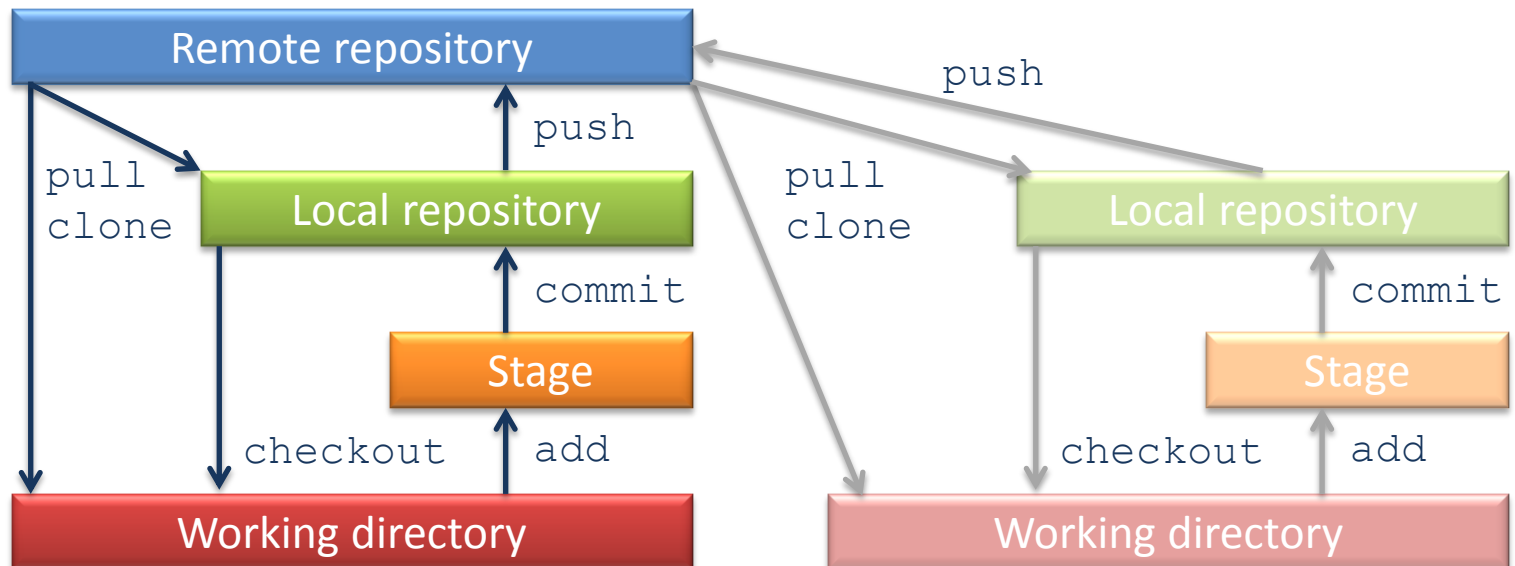
Git

Dr. Heidi E.L. Lischer
University of Zurich
Switzerland

12 June, 2015






Remote repositories

- Version control becomes really powerful when you begin to collaborate with other people
 - Manuscripts
 - Code
- Git allow us to move work between any two repositories
 - One copy on a central hub → on the web



Remote repositories

- Git hosting comparison:

Site	Price	# of repos	Type	Users	Storage
	\$0	Unlimited	Public	Unlimited	1 GB
	\$7	5	Public/private	Unlimited	1 GB
	\$0	Unlimited	Public/private	5	1 GB
	\$0	1	Private	2	1 GB
	\$0	Unlimited	Private	2	10 GB
	\$ 5	Unlimited	Private	Unlimited	500 MB

... and many more

Create GitHub repository

- Log in to GitHub

The screenshot shows the GitHub 'Create repository' page. The user 'HeidiLischer' is logged in. The 'Repository name' field is highlighted with an orange box and contains 'URPPTutorialGit' with a green checkmark. The 'Owner' dropdown is set to 'HeidiLischer'. The 'Description' field is empty. The 'Public' radio button is selected. The 'Initialize this repository with a README' checkbox is checked. The 'Add .gitignore' dropdown is set to 'None'. The 'Add a license' dropdown is set to 'None'. A 'Quick setup' modal is open at the bottom, showing the 'Set up in Desktop' button and the HTTPS URL 'https://github.com/HeidiLischer/URPPTutorialGit.git'. A blue arrow points from the 'Create repository' button to the 'Set up in Desktop' button.

Search GitHub

Explore Gist Blog Help

HeidiLischer +

New repository

New organization

Owner

HeidiLischer /

Repository name

URPPTutorialGit ✓

Great repository names are short and memorable. Need inspiration? How about [freezing-octo-robot](#).

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None

Create repository

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/HeidiLischer/URPPTutorialGit.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Connect GitHub repository

- Next step: **connect** local (already exists) with remote repository

```
git remote add origin https://github.com/HeidiLischer/URPPTutorialGit.git
```



Local nickname for the
remote repository

- **Send changes** from the local repository to the remote repository

```
git push origin master
```

- **Update local repository** to the newest changes from the remote repository

```
git pull origin master
```

Collaborating

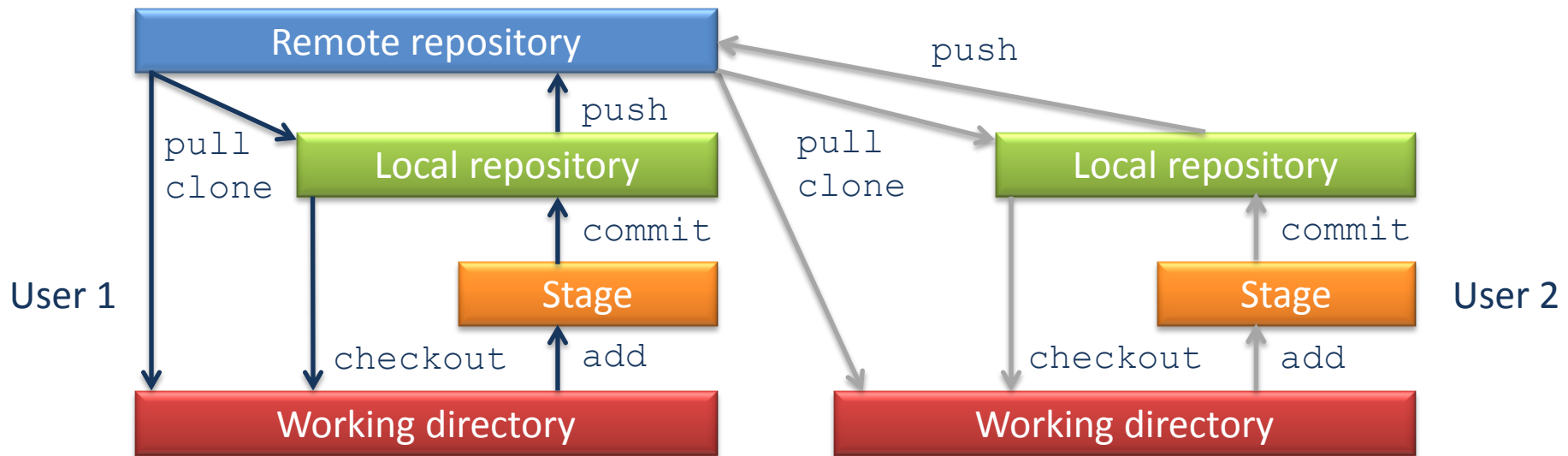
- Collaborators (or you on another computer) can have their own **copy of the repository**
 1. Give another person access:

The screenshot shows the GitHub interface for the repository **HeidiLischer / URPPTutorialGit**. The left sidebar contains a menu with **Options**, **Collaborators**, **Webhooks & Services**, and **Deploy keys**. The **Collaborators** tab is active, displaying a message: "This repository doesn't have any collaborators yet. Use the form below to add a collaborator." Below the message is a search bar labeled "Search by username or full name" and an "Add collaborator" button. The right sidebar contains icons for repository management, with the **Settings** icon (a wrench) highlighted by an orange box. The word "settings" is written in the bottom right corner.

Collaborating

- Collaborator has to **copy the repository**:

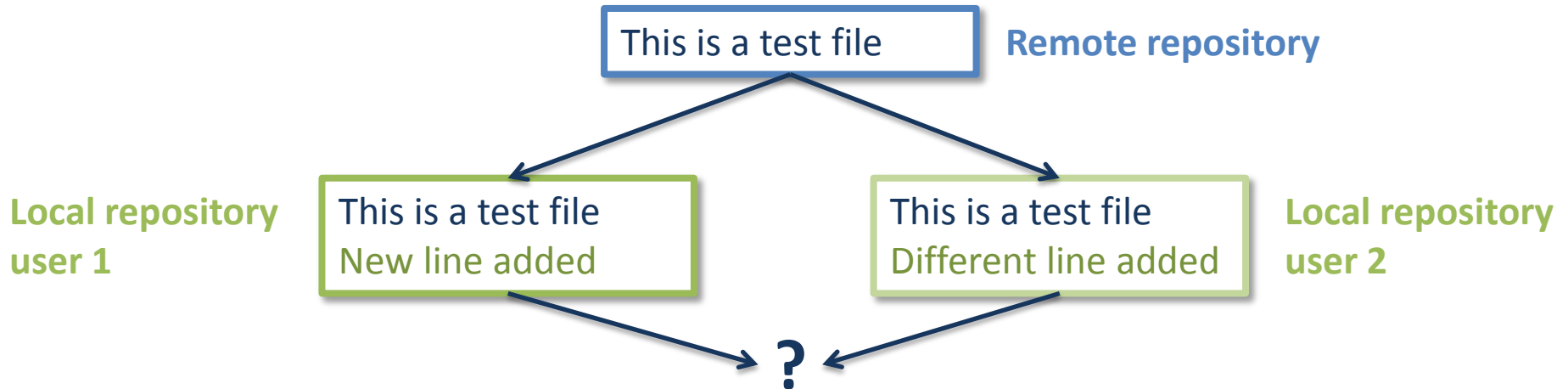
```
git clone https://github.com/HeidiLischer/URPPTutorialGit.git
```



- Collaborator can now
 - make changes in the copy of the repository: `add, commit`
 - send changes to remote repository: `push`
 - Download changes from the remote repository: `pull`

Conflicts

- Collaborators can work in parallel
→ make different changes to each copy



- Git will return an error if one try to push

```
git push origin master
! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to
'https://github.com/HeidiLischer/URPPTutorialGit.git'
...
```


Conflicts

- **Resolve conflicts:**
 1. **Pull changes** from remote repository

```
git pull origin master
...
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

→ pull tells us that there is a conflict

→ marks conflict in the affected file

test.txt:

```
This is a test file
<<<<<<< HEAD
Different line added
=====
New line added
>>>>>>> dabb4c8c450e8475aee9b14b4383acc99f42af1d
```

What user 2 tries
to add

What user 1
already added

Conflicts

2. Merge changes into the copy we are working in

- keep change made in local repository
- keep change made in remote repository
- something new to replace both
- ...

```
git add test.txt  
git commit -m "merging changes"
```

3. Push it

```
git push origin master
```

- git keeps track of what one as merged
- partner doesn't need to merge again if he pulls changes

Licensing

- Public remote repository with source code, manuscripts or other creative work
→ you should think about licensing
- How to choose an appropriate license?
 - Many possibilities
 - Morin, Urban and Sliz; PLoS Computational Biology 2012:
“A Quick Guide to Software Licensing for the Scientist-Programmer”
- **Most popular** open source licenses:
 - GNU General Public License (GPL)
 - MIT license
 - BSD license

Acknowledgment

- Sources:
 - <http://swcarpentry.github.io/git-novice>
 - <http://rogerdudler.github.io/git-guide/>