University of Zurich[UZH]

URPP Evolution in Action

URPP tutorial

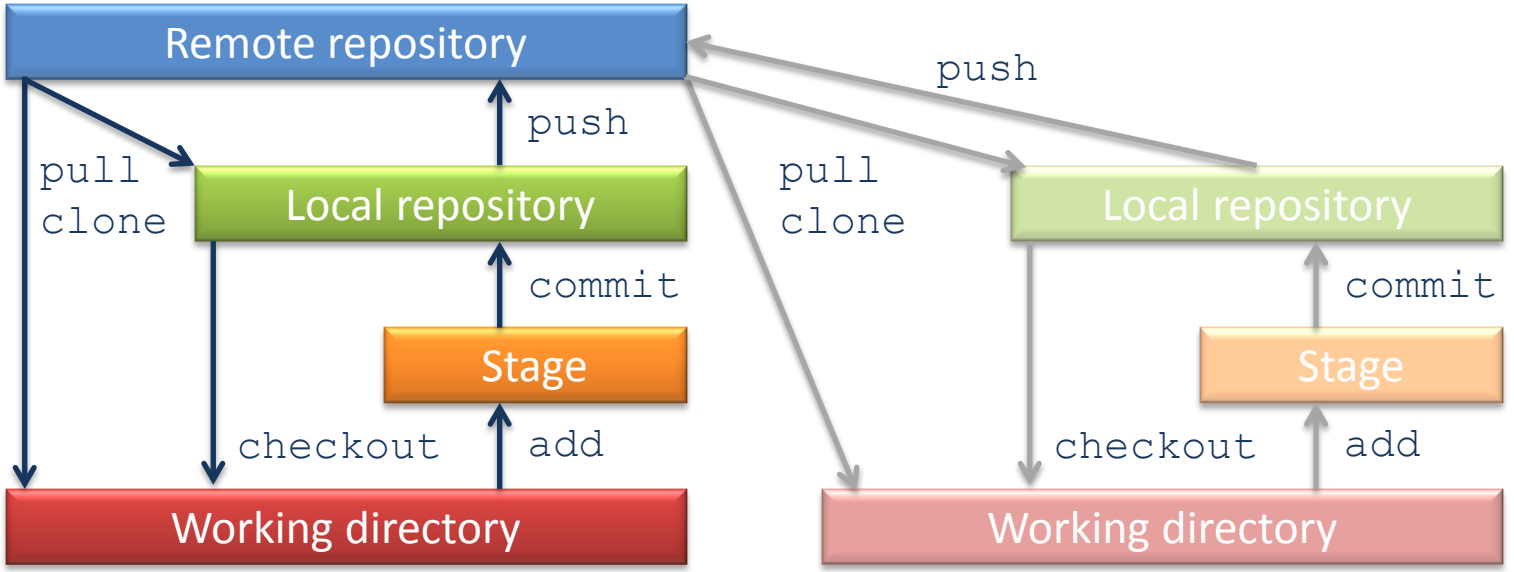
Git

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

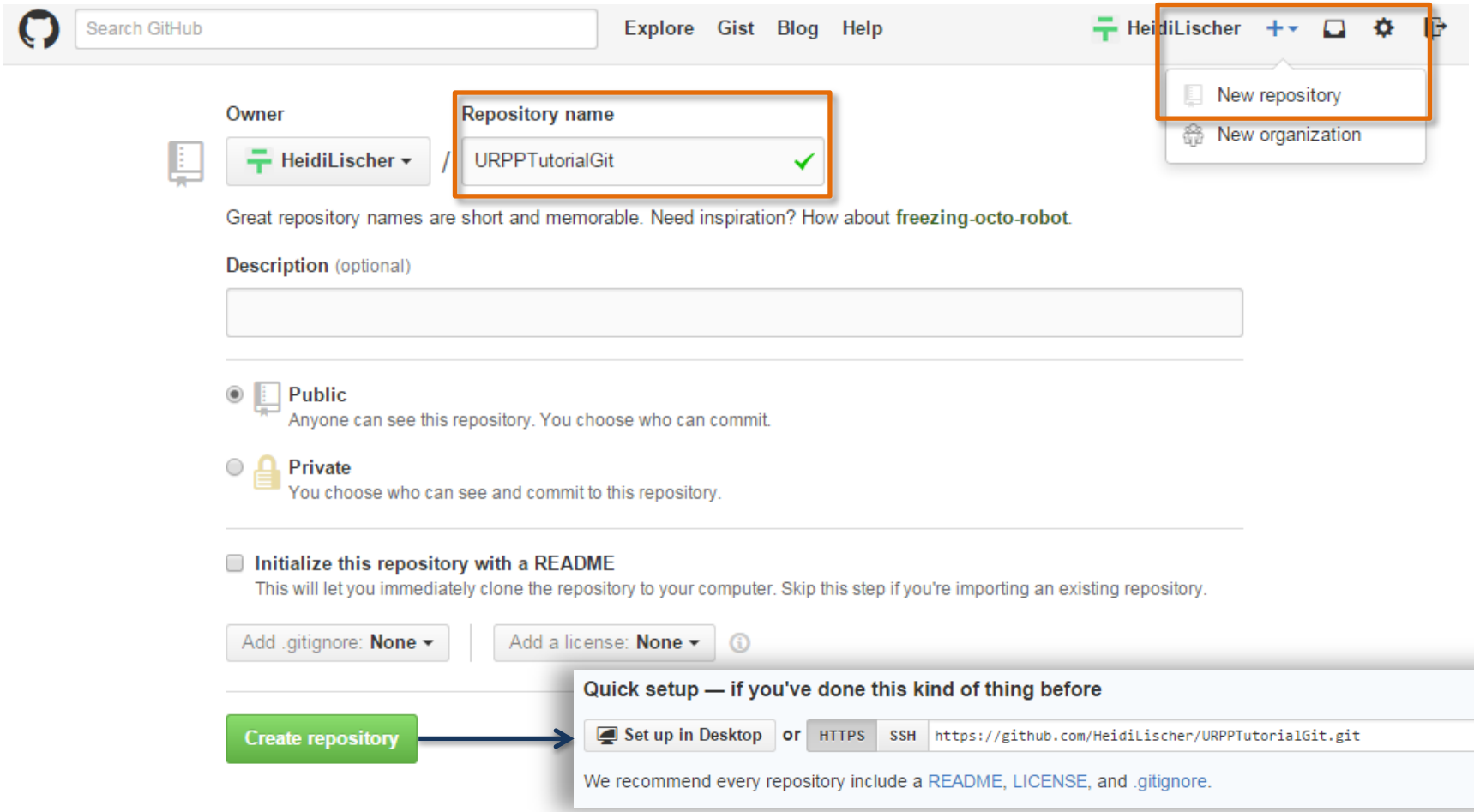
- Git hosting comparison:

Site	Price	# of repos	Type	Users	Storage
GitHub	$0	Unlimited	Public	Unlimited	1 GB
	$7	5	Public/private	Unlimited	1 GB
Atlassian Bitbucket	$0	Unlimited	Public/private	5	1 GB
assembla	$0	1	Private	2	1 GB
Kiln	$0	Unlimited	Private	2	10 GB
GITGO	$ 5	Unlimited	Private	Unlimited	500 MB

… and many more

Create GitHub repository

- Log in to GitHub



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

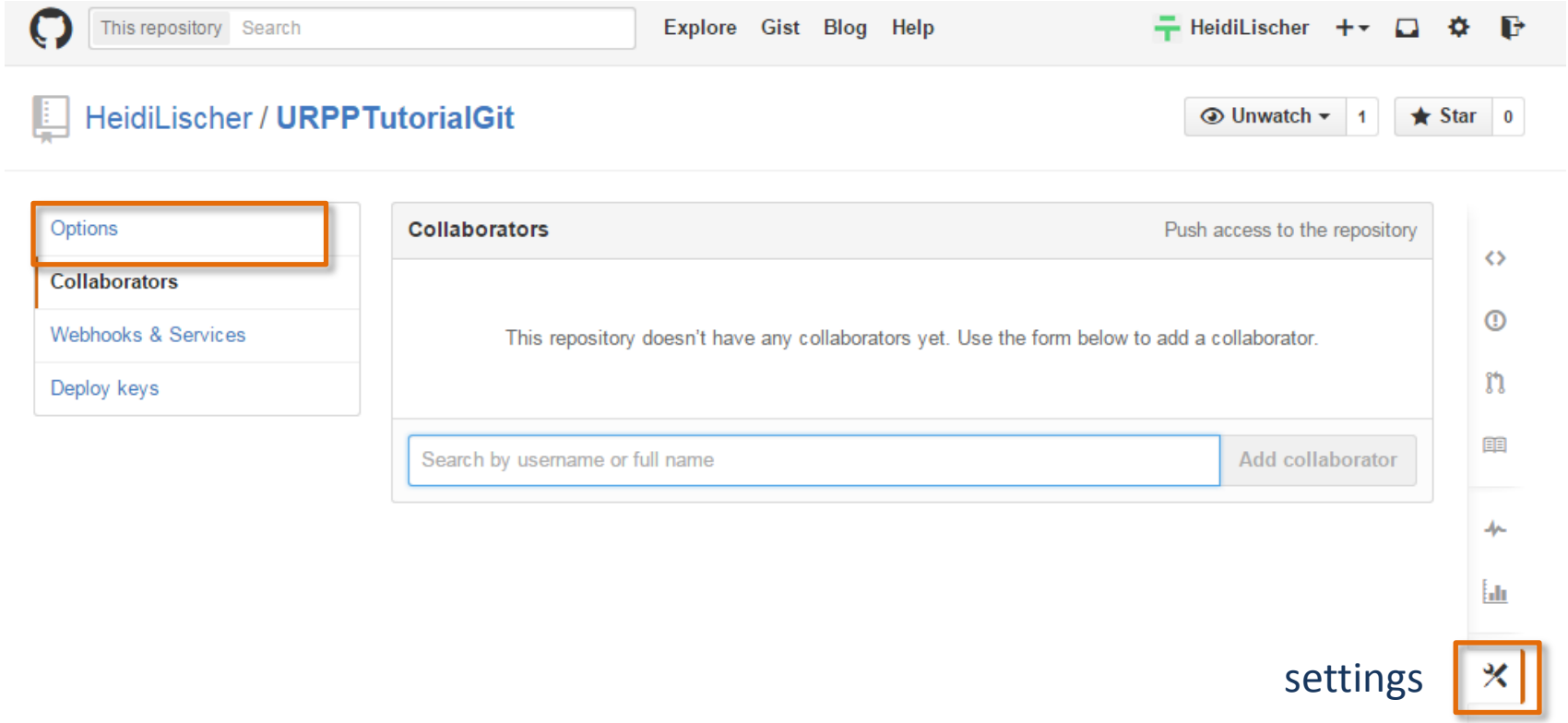
```
git push origin master
```

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

```
git pull origin master
```

Collaborating

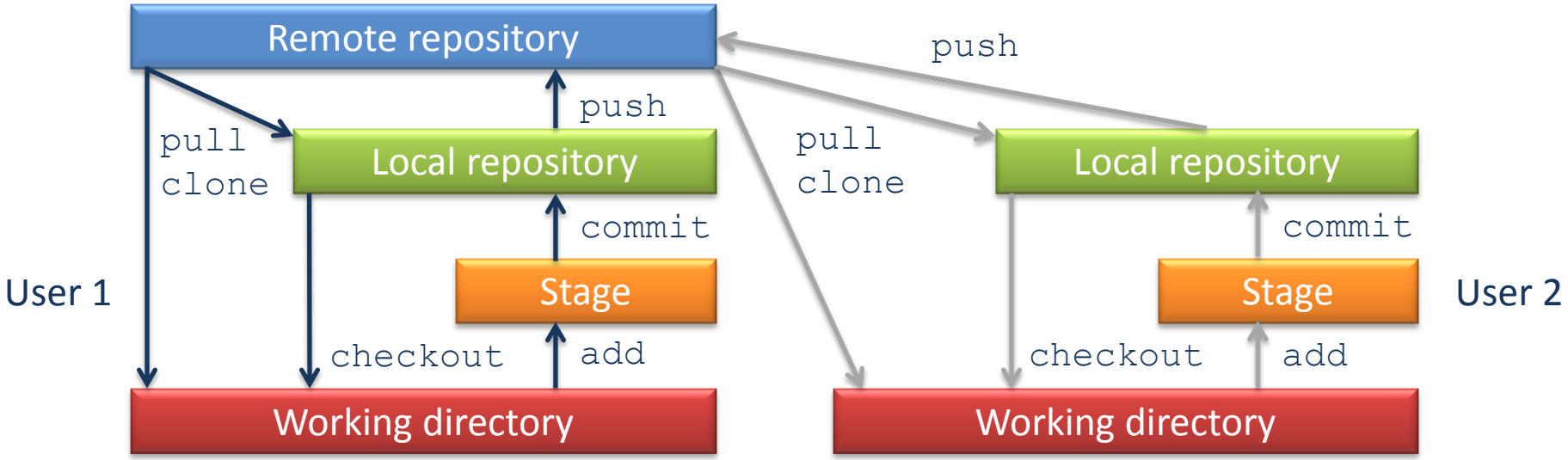
- Collaborators (or you on another computer) can have their own copy of the repository
  1. Give another person access:



Collaborating

- Collaborator has to copy the repository:

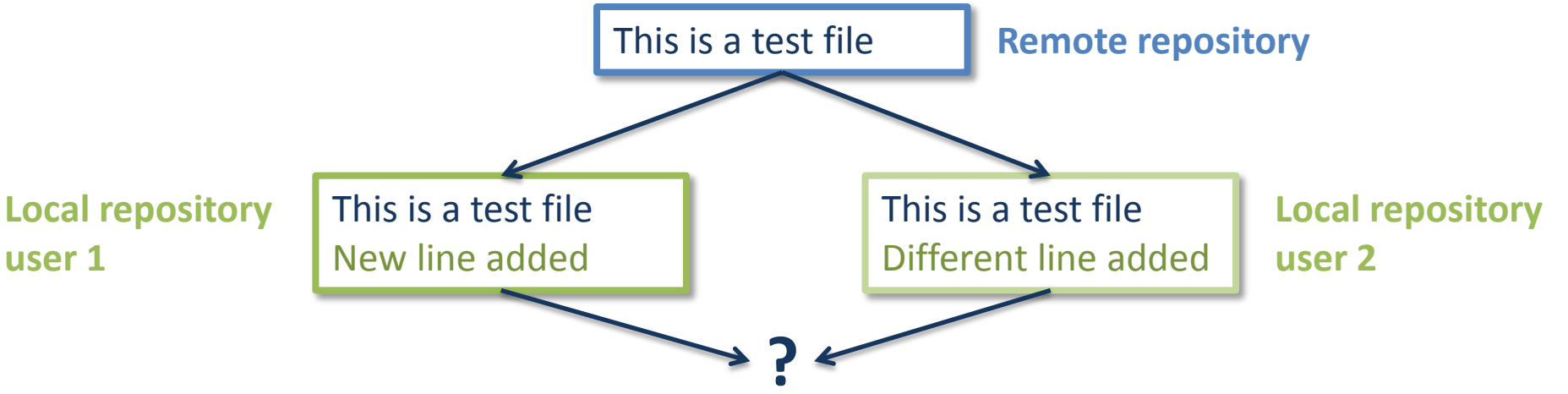
```
git clone https://github.com/HeidiLischer/URPPTutorialGit.git
```



- 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/