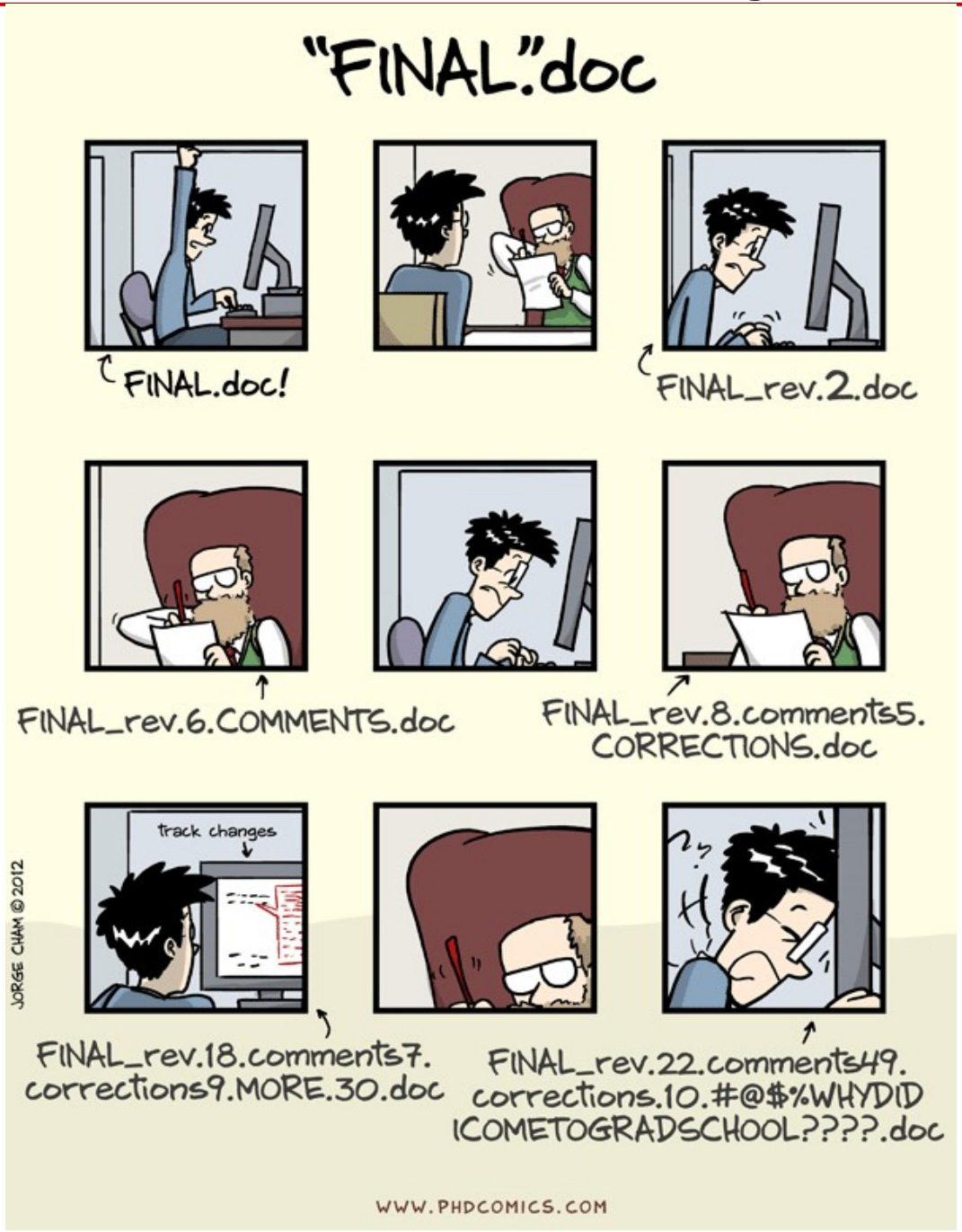
Version Control System
GIT

Version Contol System

- Version (revision) control systems are software that help you track changes you make in your code over time. As you edit to your code, you tell the version control system to take a snapshot of your files. The version control system saves that snapshot permanently so you can recall it later if you need it.
- Automates the storing, retrieval, logging, identification, and merging of revisions (versions).

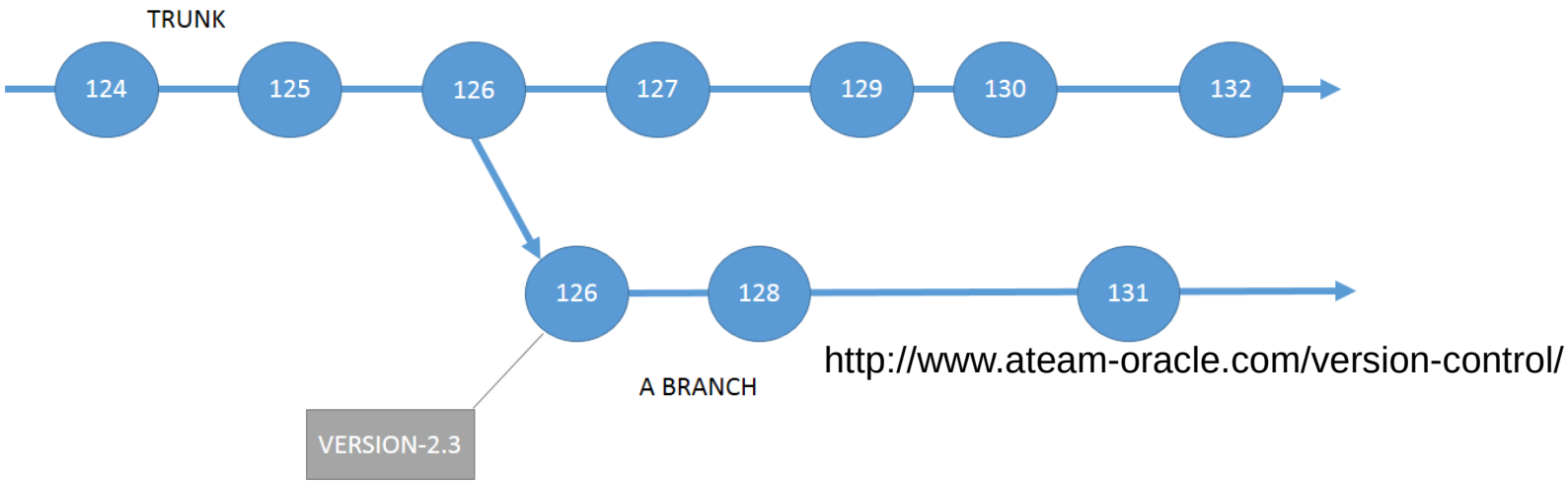
Version Contol System

Version Contol System

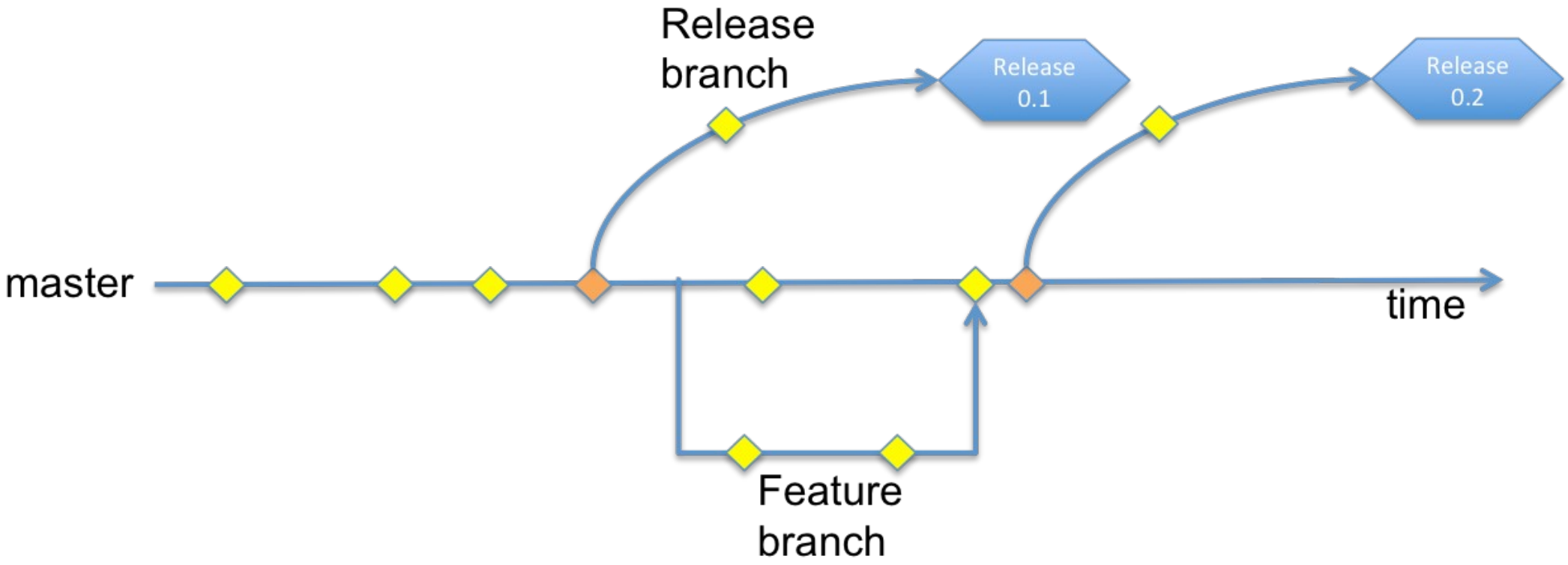


Version Contol System

- Version Control helps with:
  - Backups
  - Reproducibility
  - Comparing arbitrary historical versions
  - Maintaning multiple live versions
  - Team coding
  - Workfows (process enforcement and permisions)



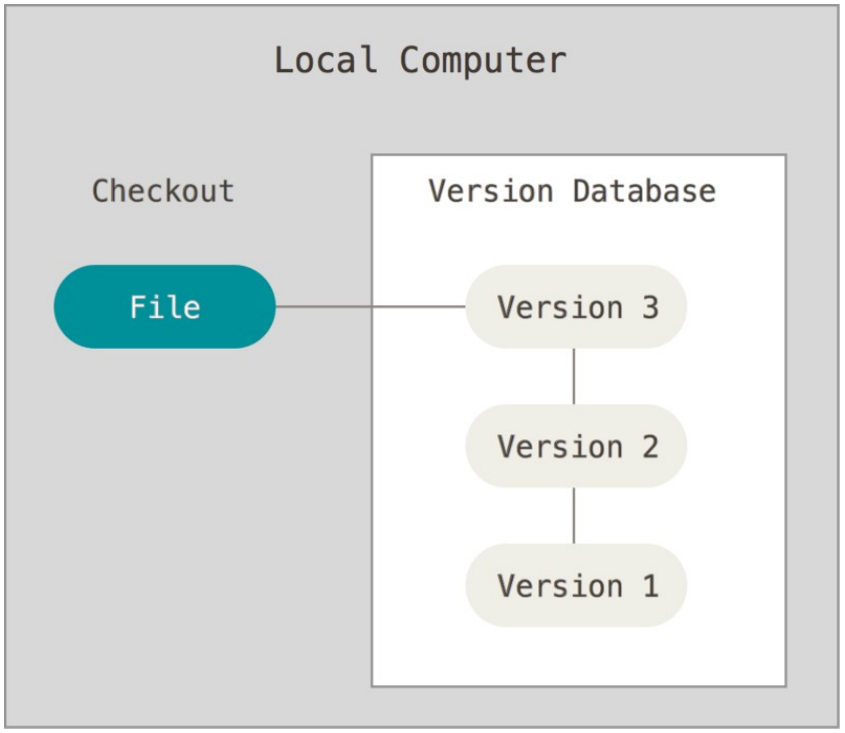
Version Contol System



http://cswsolutions.com/blog/git-non-developers-part-ii/

Version Control

- Local Version Control

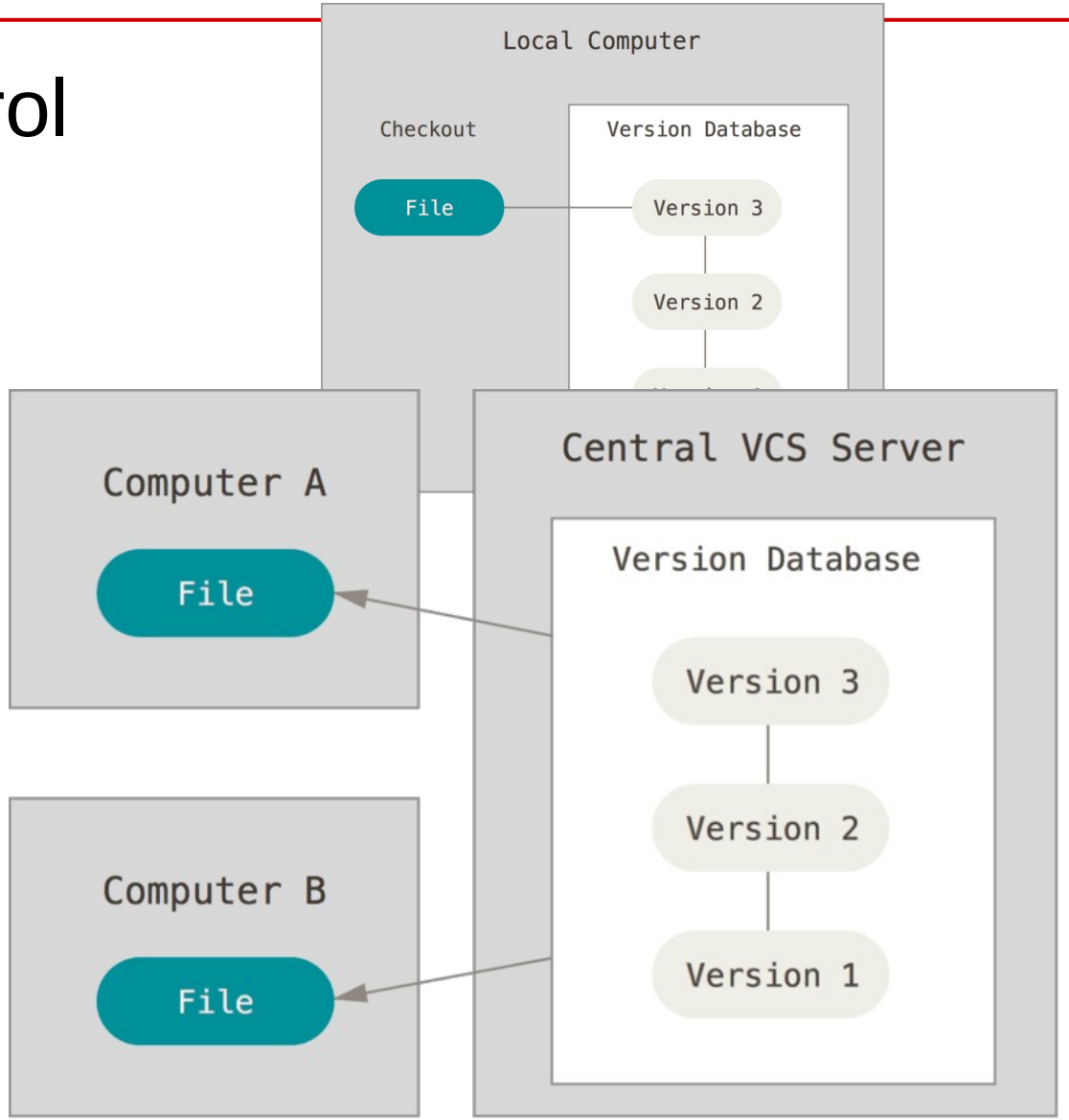


https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Version Control

- Local Version Control

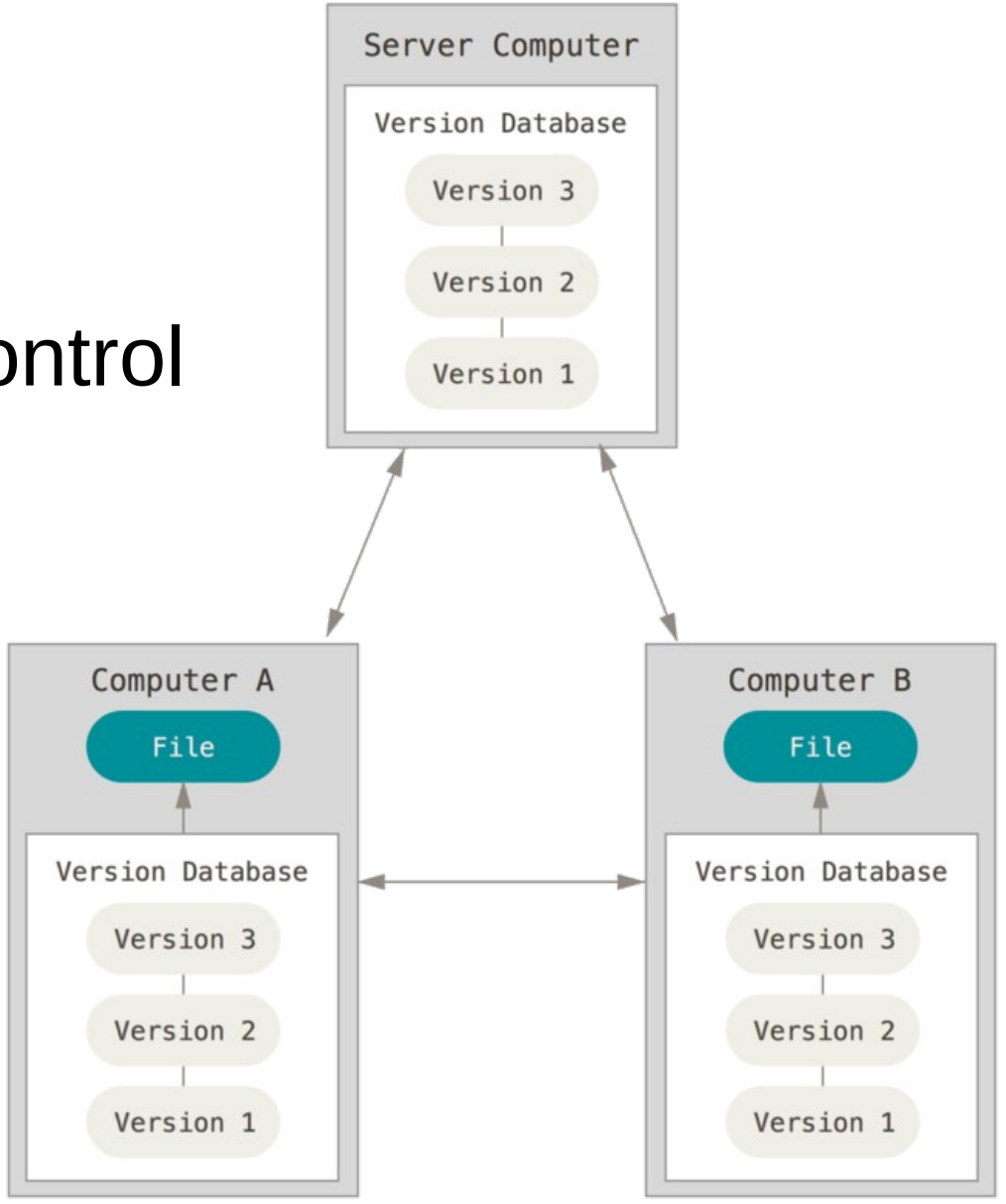
- Centralized Version Control
  - Subversion (SVN)



https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Version Control

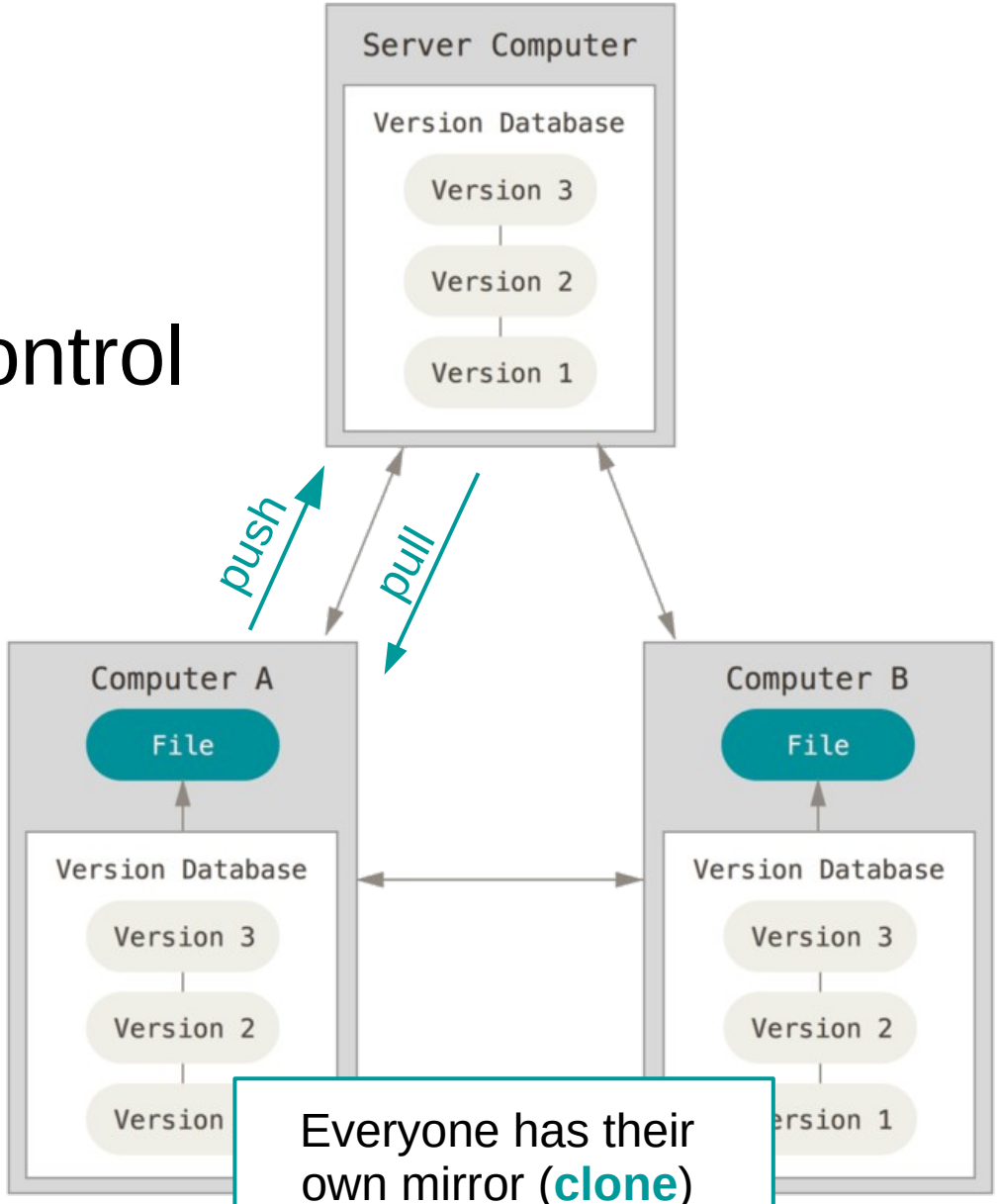
- Local Version Control
- Centralized Version Control
  - Subversion (SVN)
- Distributed Version Control
  - GIT



https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Version Control

- Local Version Control
- Centralized Version Control
  - Subversion (SVN)
- Distributed Version Control
  - GIT

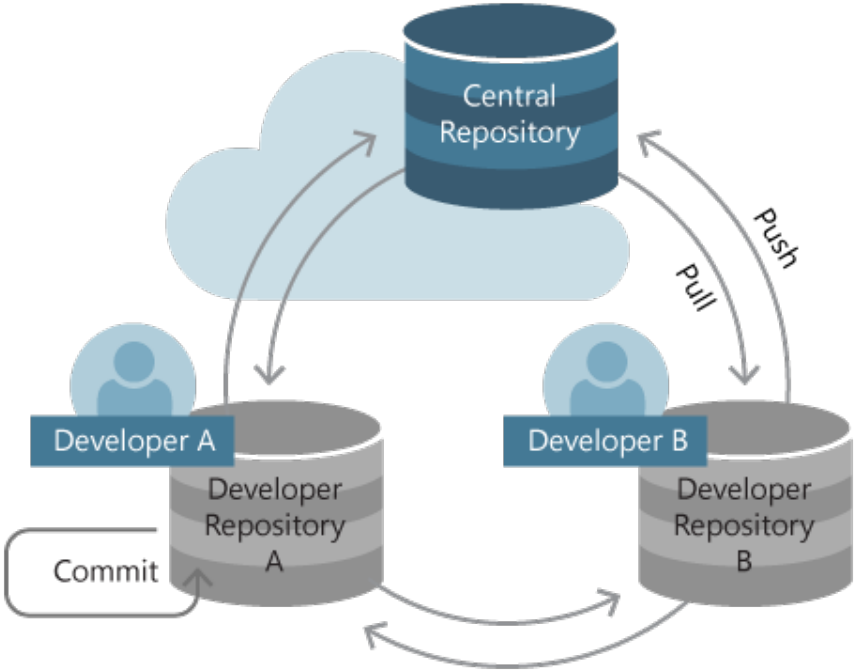


GIT in Visual Studio

https://www.visualstudio.com/learn-git/

Creating GIT repository

- A Git repository, or repo, is a folder that you've told Git to help you track file changes in.
- A Git repo contains every version of every file saved in the repo. Git saves these files very efficiently, so having a large number of versions doesn't mean that it uses a lot of disk space.



https://www.visualstudio.com/en-us/docs/git/tutorial/creatingrepo
https://www.visualstudio.com/learn/set-up-a-git-repository/

Creating GIT repository

1) From a new solution

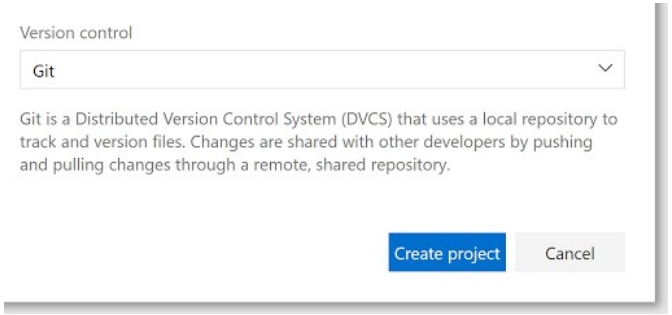
While creating new solution check appropriate checkbox

2) From an existing solution

To create a repo from an existing solution, select the Add to Source Control button in the bottom-right of the lower status bar Visual Studio publish button. This creates a new Git repo in the same directory as your solution and opens up the Publish view in Team Explorer so you can push your code to Visual Studio Team Services or another remote Git repository.

3) Create repo in Team Services

Create New project.
See next slides.

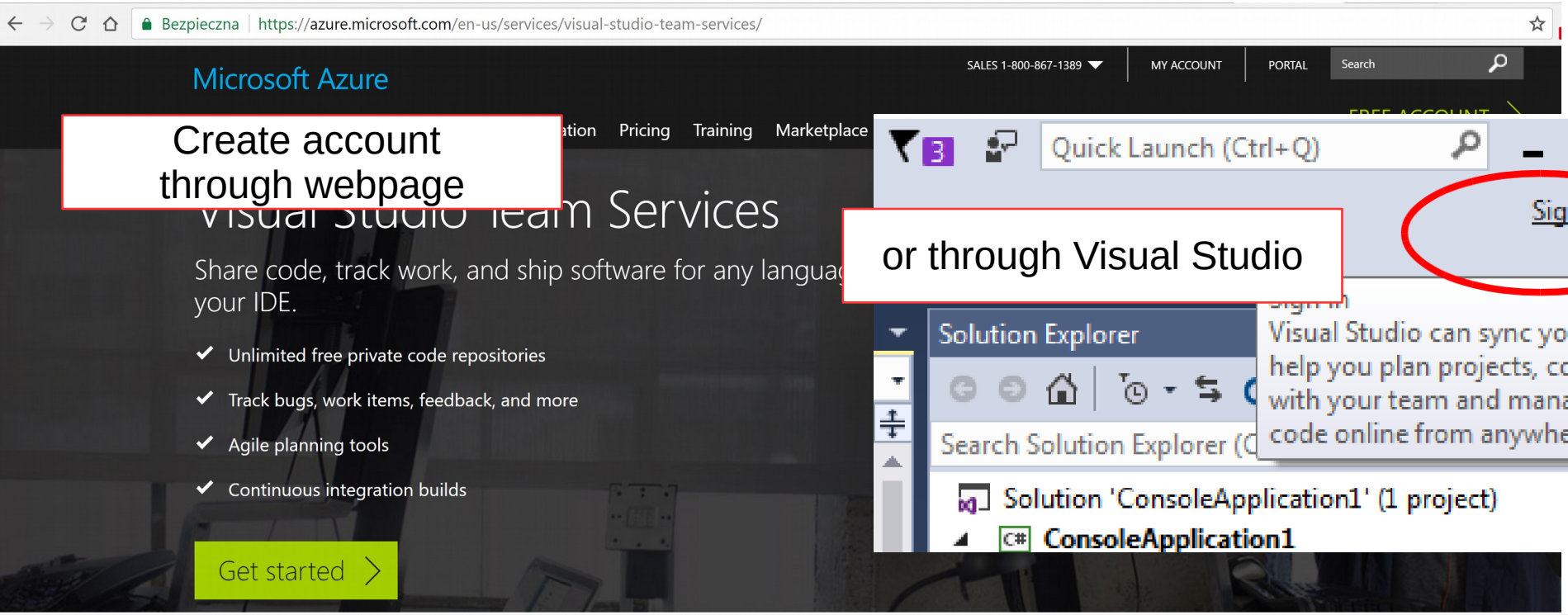


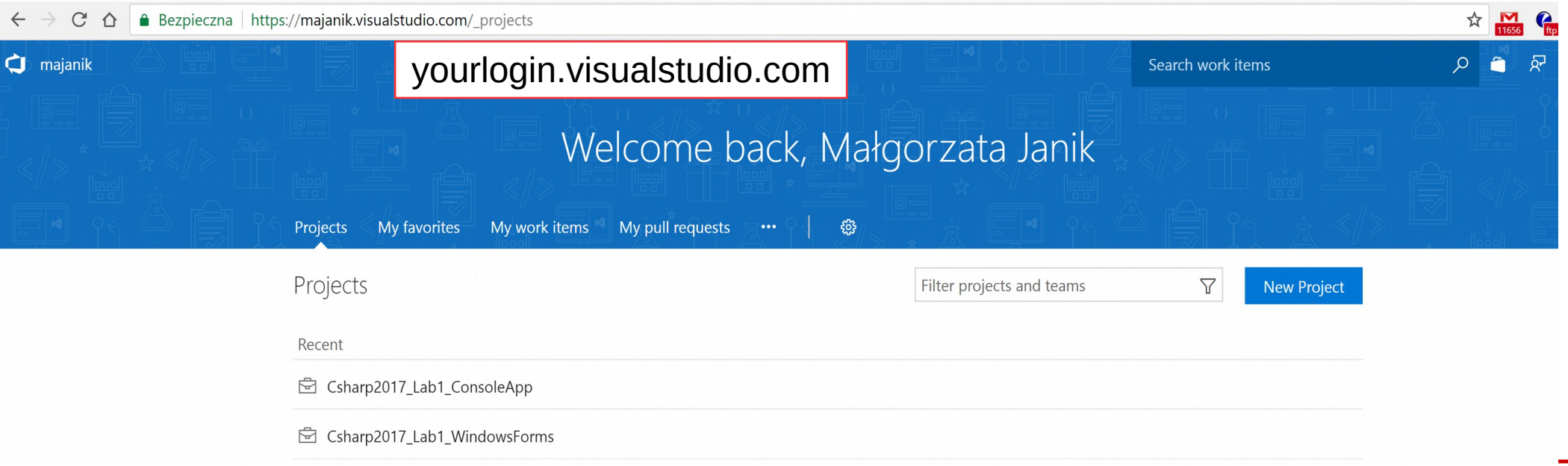
4) In an empty folder

5) Connect local repo to a remote

https://www.visualstudio.com/en-us/docs/git/tutorial/creatingrepo
https://www.visualstudio.com/learn/set-up-a-git-repository/

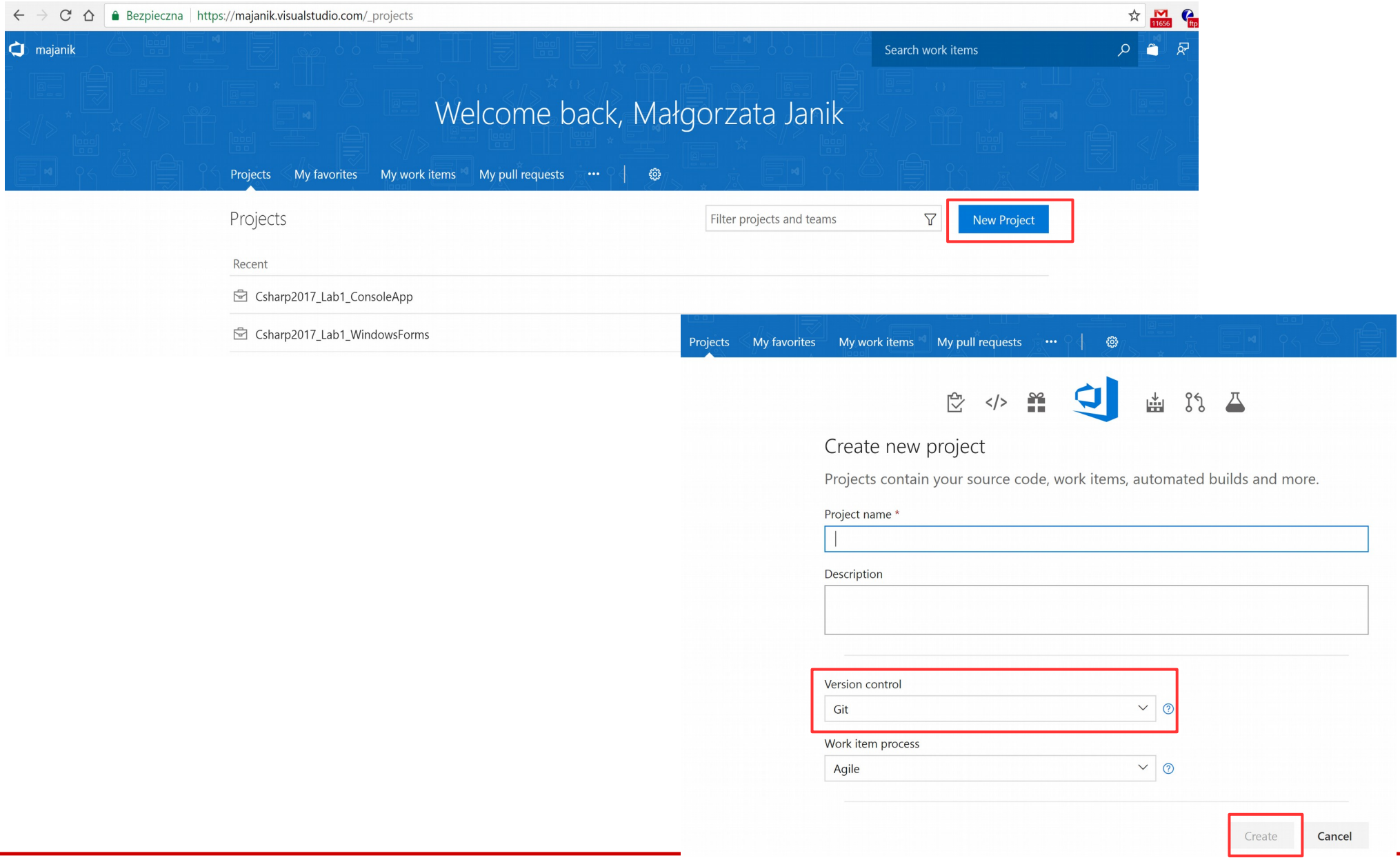
Team Services





Create repo in Team Services

- Log into Team Services and create New Project

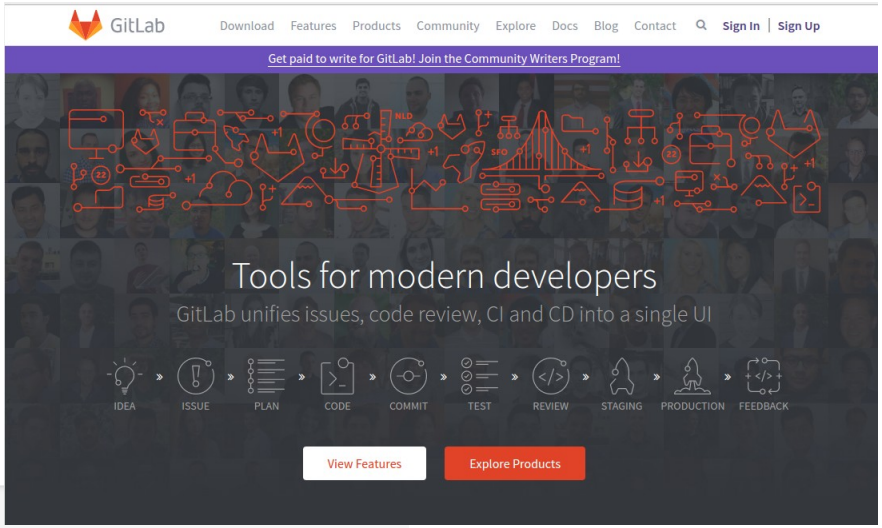


GIT Task

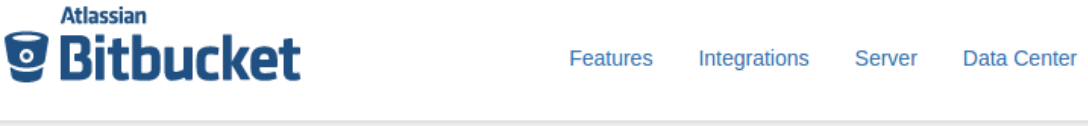
- Using Team Services web account create new GIT repository for todays project
  - You can also create a solution (**Console Application**)
- Create local copy of the repository (Clone) using Team Explorer in Visual Studio (see next slides)
- Apply some changes (ask user to press key to continue), commit them, and push to the repository (see next slides)

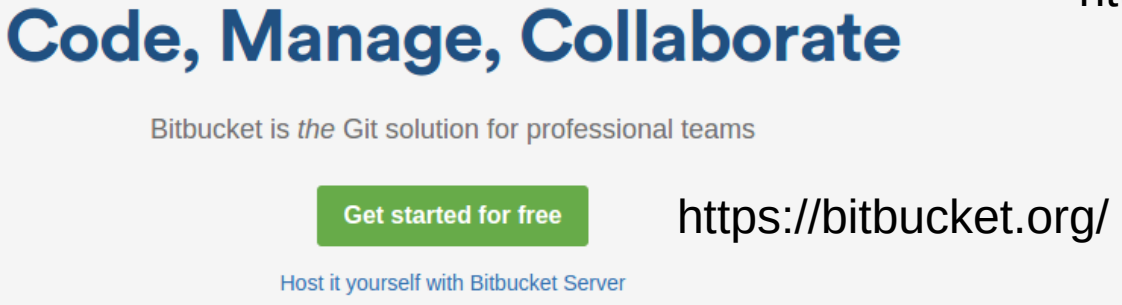
Git repository hosting services

https://github.com/



https://gitlab.com/

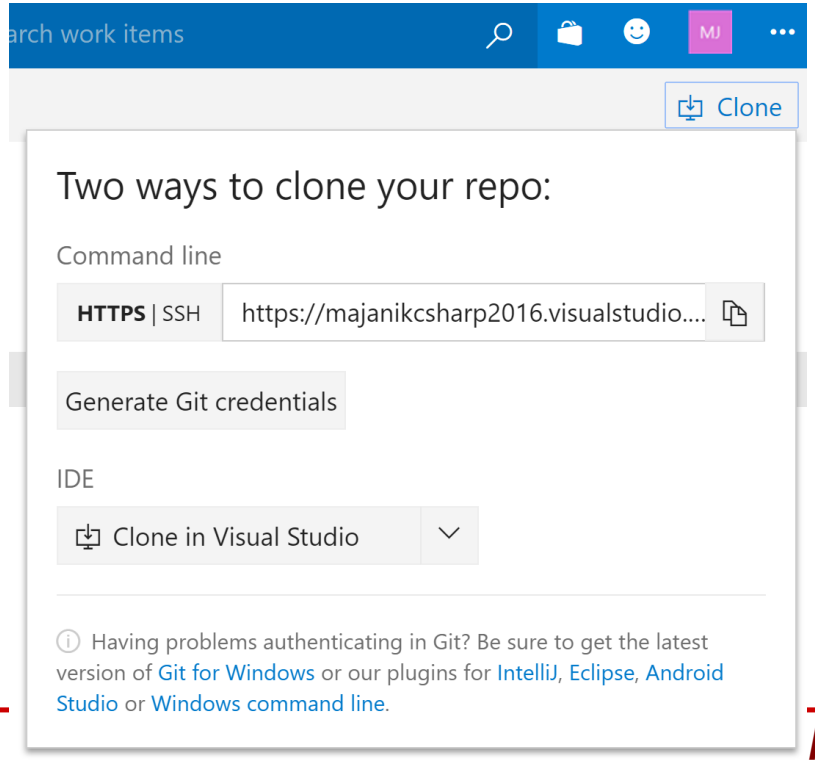




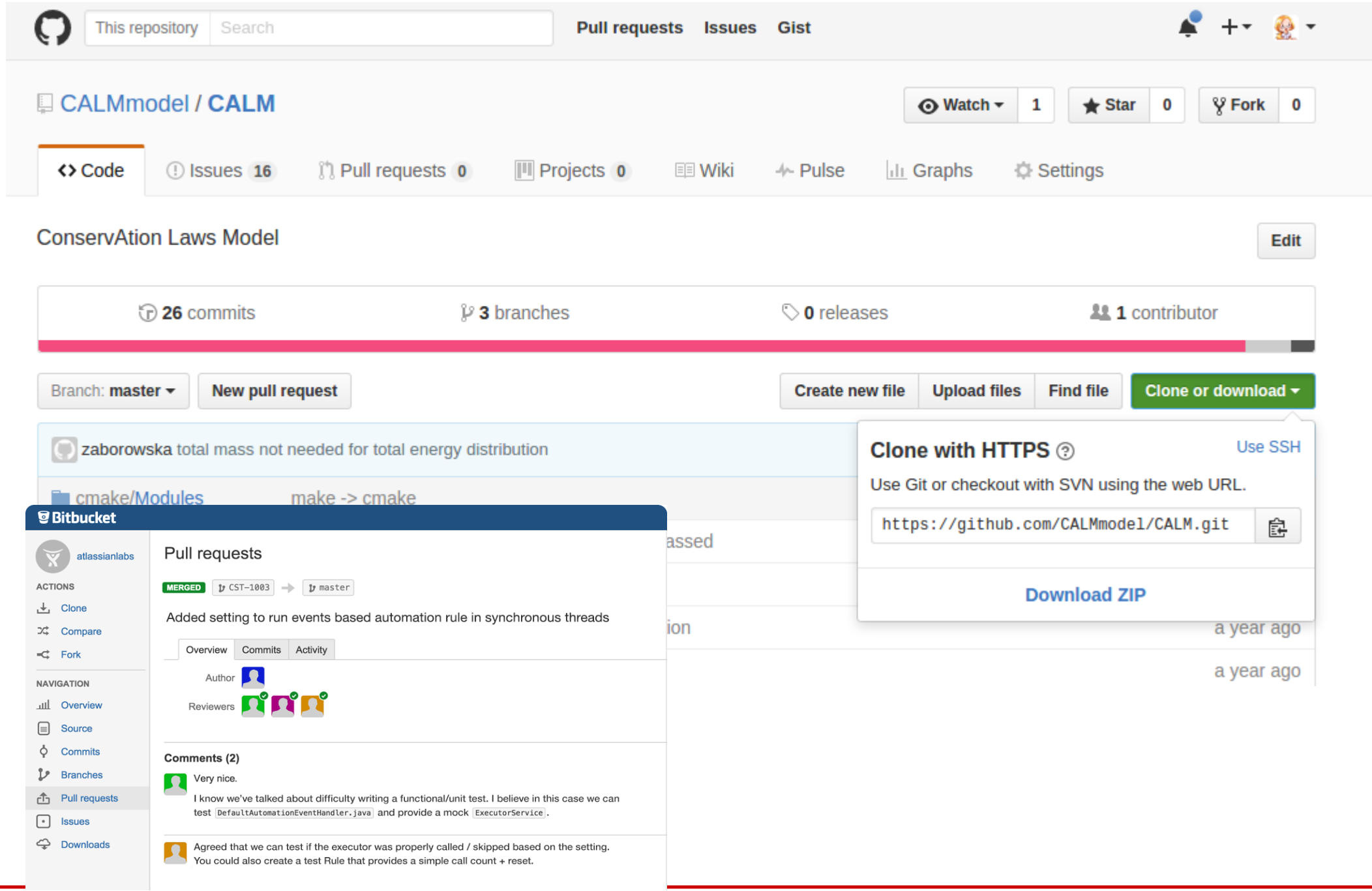
https://bitbucket.org/

Creating local repository

- Create a complete local copy of an existing Git repo by cloning it.
- 1st step: obtain URL of the source of your repo
  - In Team Services in can be found via web interface
  - GitHug / other git providers: similar, there will be „Clone” or „Download” button

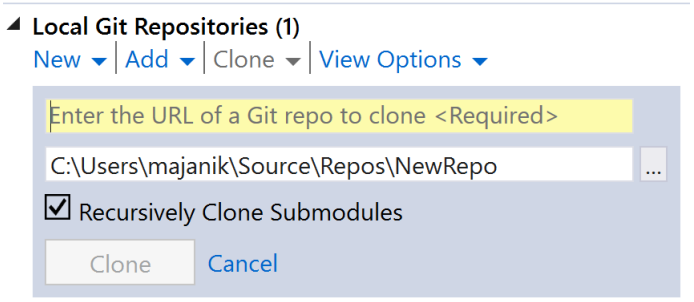


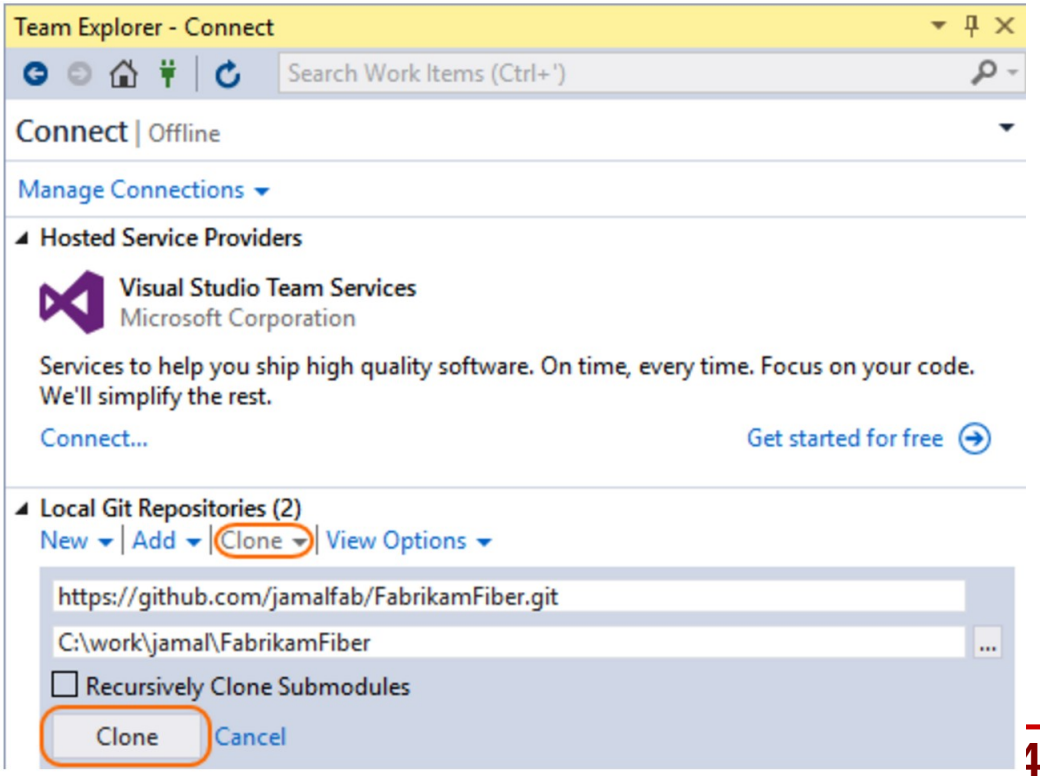
Clone URL


This repository
Search
Pull requests
Issues
Gist
CALMmodel / CALM
Watch 1
Star 0
Fork 0
Code
Issues 16
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
ConservAtion Laws Model
Edit
26 commits
3 branches
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
zaborowska total mass not needed for total energy distribution
cmake/Modules
make -> cmake
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/CALMmodel/CALM.git
Download ZIP
a year ago
a year ago
Bitbucket
atlassianlabs
ACTIONS
Clone
Compare
Fork
NAVIGATION
Overview
Source
Commits
Branches
Pull requests
Issues
Downloads
Pull requests
MERGED
CST-1003
master
Added setting to run events based automation rule in synchronous threads
Overview
Commits
Activity
Author
Reviewers
Comments (2)
Very nice.
I know we've talked about difficulty writing a functional/unit test. I believe in this case we can test DefaultAutomationEventHandler.java and provide a mock ExecutorService.
Agreed that we can test if the executor was properly called / skipped based on the setting.
You could also create a test Rule that provides a simple call count + reset.

2nd step: clone the repo

- In Team Explorer, open the Connect view.
- Select Clone under Local Git Repositories and enter the URL.
- Select a folder where you want your cloned repo to be kept.
- Select Clone to clone the repo

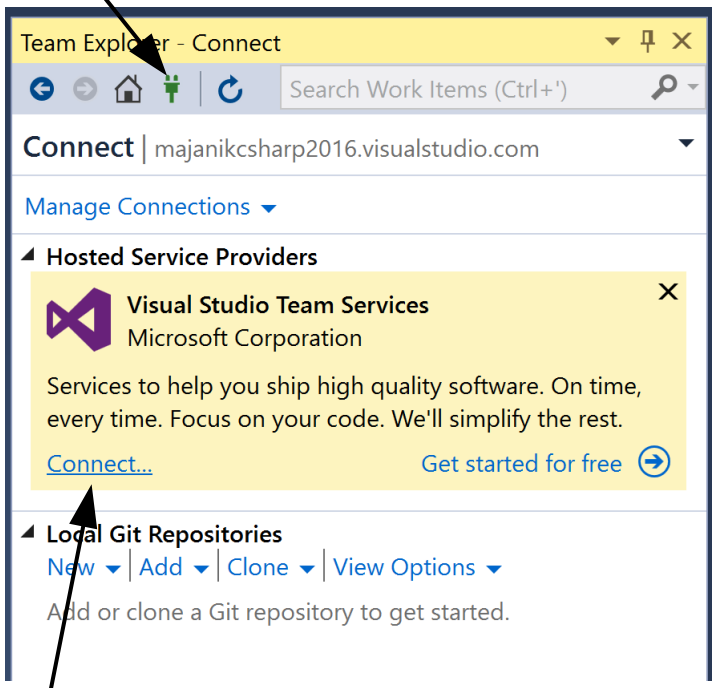




If you use VS Team Services...

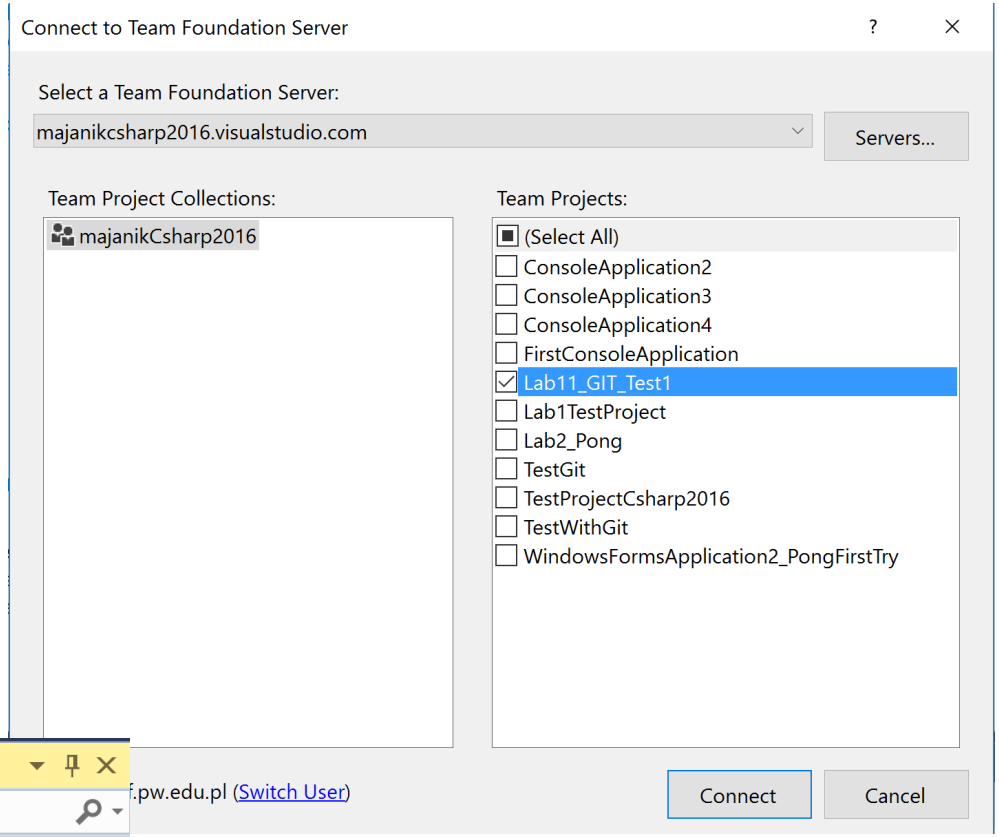
Manage Conections

(1)

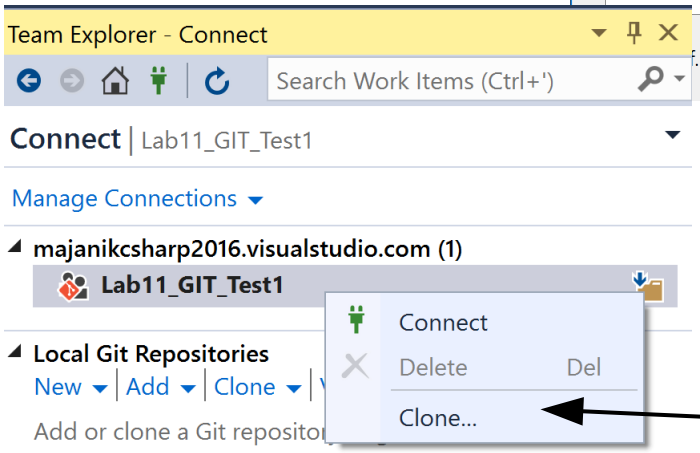


Connect

(2)



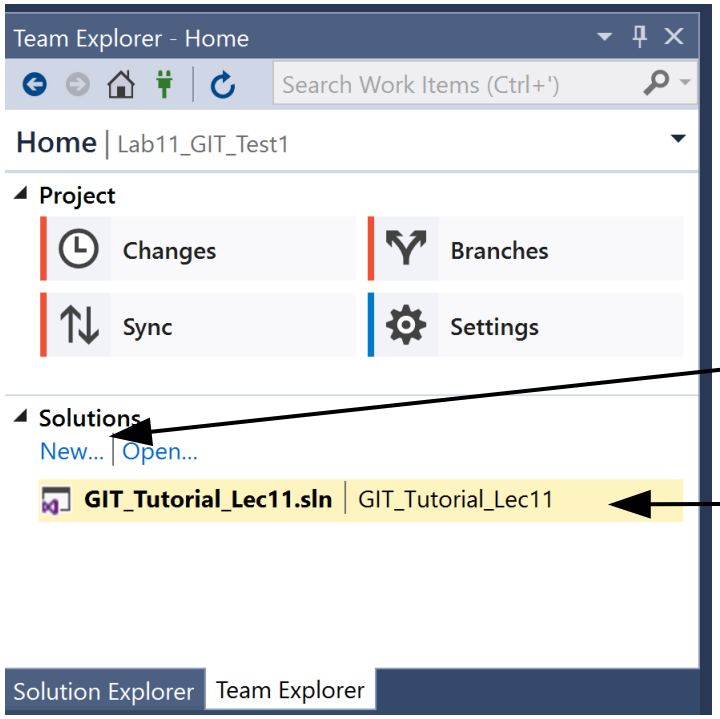
(3)



Clone

Open Cloned solution

- When the solution is cloned (you have a local copy) you can just double-click it from the Team Explorer

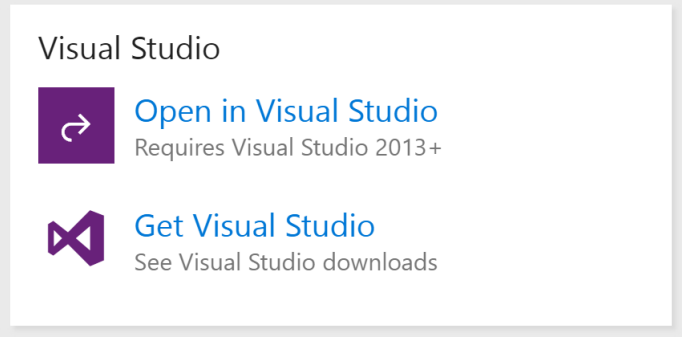


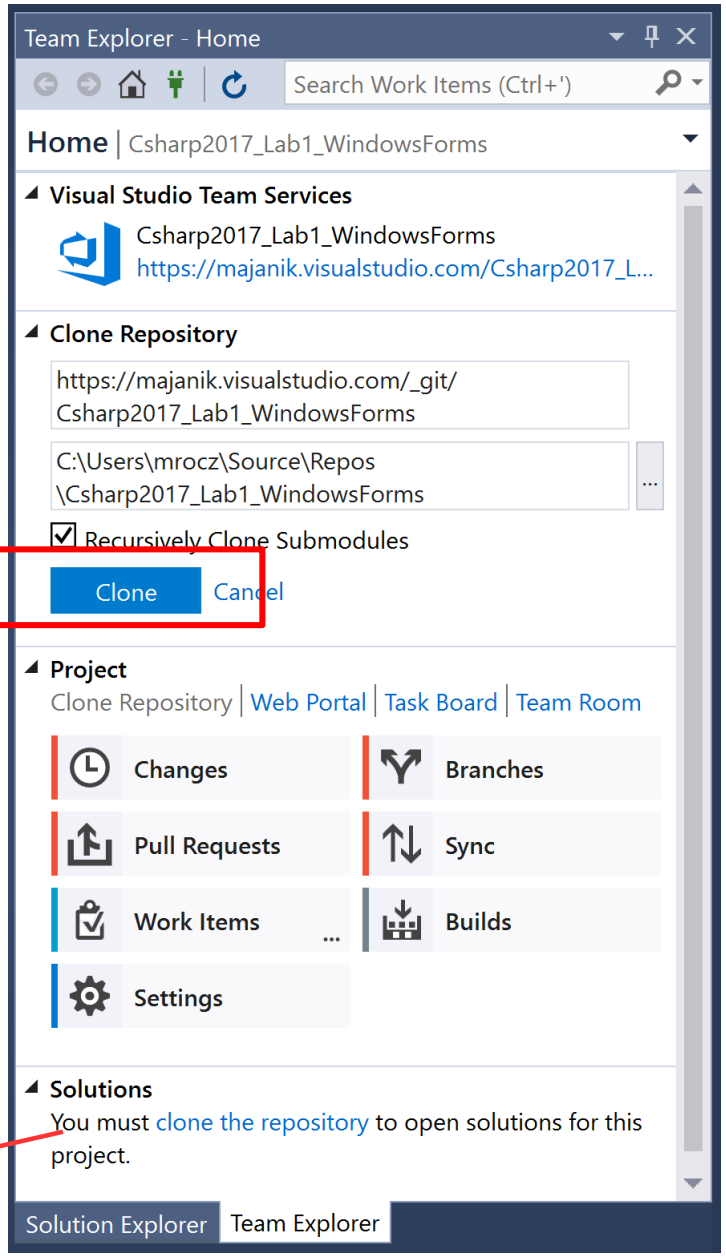
Create new solution if there is nothing on the list

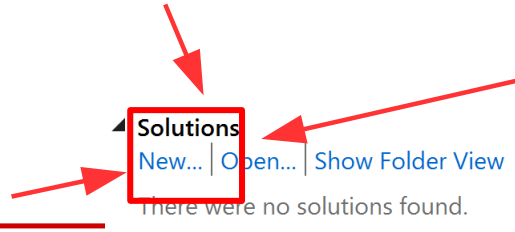
Open solution from git

Alternatively….

- create project in TS
- synchronize it with local repository
- as on Lab 1







Updating the repository

– sharing your code

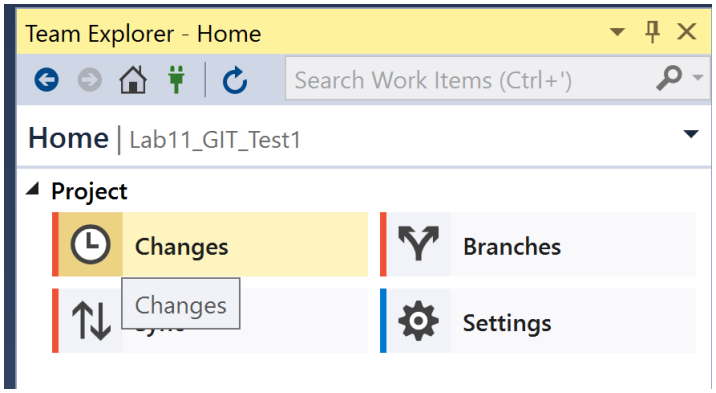
- **Stage changes** (show git which files you would like to update)
- Create a **commit** (save staged files to your repo)
- **Push** code to the repository (update the remote repository = share your code with others)

Commits

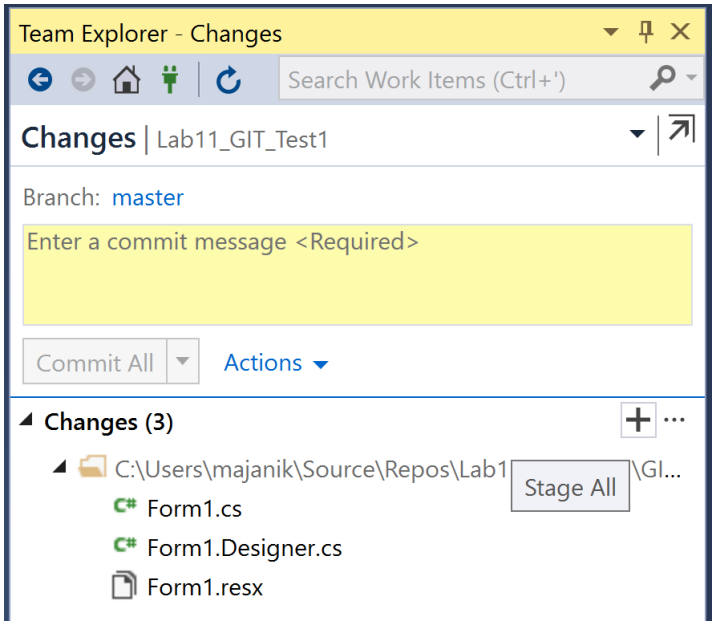
- What is a commit?
  - Commit saves changes to your local git repo
- Commit consists of:
  - The file(s) changed in the commit (**staged files)**
  - Message describing commit (important!)
  - Reference to previous commit

Commits via Team Explorer

- Open „Changes” view in Team Explorer



- Stage changes
  (possibility to select specific files)



- Write a commit message and ***Commit All*** / *Stage Commit*

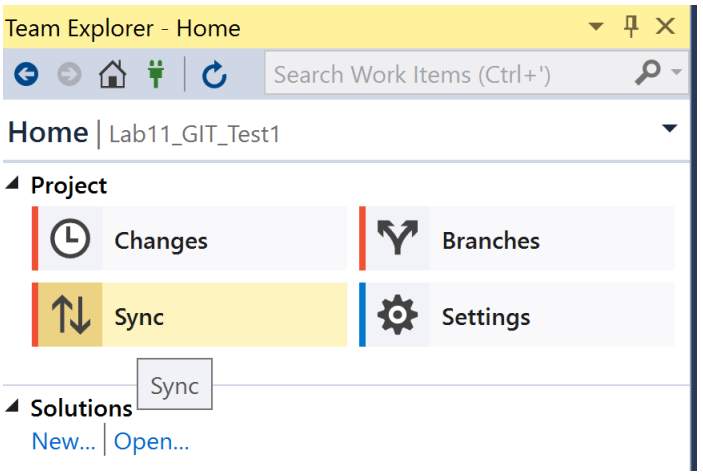
- If needed, it is possible to update your last commit to correct small errors without making a new commit.
  Action → Amend Previous Commit.

Share code with push

- You can share the changes in commits using **push** command
- This command uploads changes to (**synchronizes**) the central repository
- Git makes sure that pushed changes are consistent with the remote branch so others can pull your commits and merge them into their own local copy

Synchronization

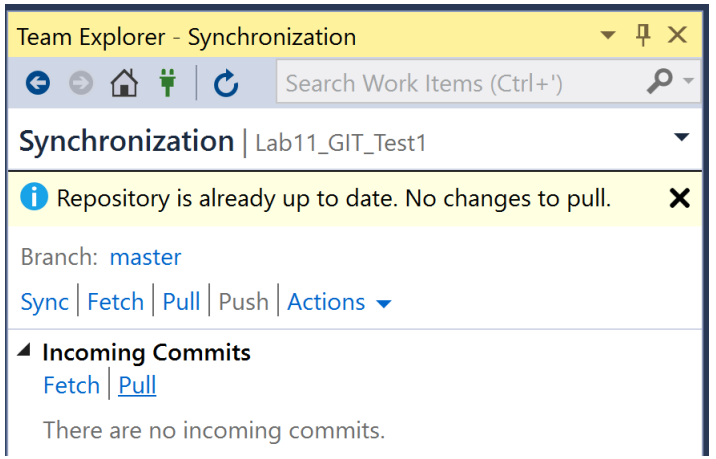
- Open Sync view in Team Explorer



- **Push** will push the commits to the remote branch.
- Resolve merge conflicts before pushing
  - If there are conflicts between your local commits on the commits on the remote branch, you must first resolve these conflicts before you can push your changes. You should pull the changes from others first, resolve the conflicts and commit the changes, then re-attempt the push.

Update local repository

- Update the code in your local repo with the changes from other members of your team using the following commands:
  - **fetch** , which downloads the changes from your remote repo but does not apply them to your code.
  - **merge** , which applies changes taken from fetch to a branch on your local repo.
  - **pull** , which is a combined command that does a fetch and then a merge.
    - In Visual Stuio:
      use Sync option in Team Explorer



https://www.visualstudio.com/en-us/docs/git/tutorial/pulling

GIT Task

- Make sure you have tested all the steps:
- Using Team Services web account create new GIT repository for todays project
  - You can also create a solution (**Console Application**)
- Create local copy of the repository (Clone) using Team Explorer in Visual Studio
- Apply some changes (ask user to press key to continue), commit them, and push to the repository

References

- GIT
  - https://www.visualstudio.com/en-us/docs/git/tutorial/gitworkflow
  - https://www.visualstudio.com/en-us/docs/git/gitquickstart

THE END

dr inż. Małgorzata Janik

malgorzata.janik@pw.edu.pl