

# Test Automation

## Lecture 16 –

### Version Control with Git



IT Learning &  
Outsourcing Center

Lector: Milen Strahinski  
Skype: strahinski  
E-mail: [milen.strahinski@pragmatic.bg](mailto:milen.strahinski@pragmatic.bg)  
Facebook: <http://www.facebook.com/LamerMan>  
LinkedIn: <http://www.linkedin.com/pub/milen-strahinski/a/553/615>

[www.pragmatic.bg](http://www.pragmatic.bg)

# Table of contents



- What is version control software?
- Why do you need it?
- Where did it come from?
- Types of VCS
- How does it work?
- Which one should I use?
- How do I start using it?



# What is version control software?



IT Learning &  
Outsourcing Center

- **Version Control Software (VCS) is a set of:**
  - programs that manage changes to computer files, such as documents, images and source code.
- Also known as:
  - Revision control software
  - Version management software
  - Source control software
  - Configuration management software

# Why do you need it?



- Undo, incremental backup of changes
- Trying out ideas
- Integrating several sources or subsystems
- Collaboration with other people
- Troubleshooting
- Statistics
- Productivity
- Sanity

# Does this look familiar?



IT Learning &  
Outsourcing Center



Project



Project.2011-10-21



Project.old



Project.2011-10-  
21\_morning



Project.1



Project.working



Project.11



Project.broken



Project.111



Project.backup

# Where did it come from?

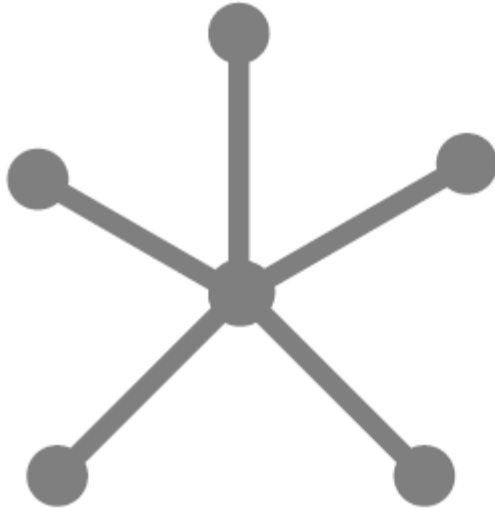


- <https://www.flourish.org/2011/12/astonishments-ten-in-the-history-of-version-control/>
- Engineering: blueprints
- Law: contract redline, legal blackline
- Software Development: early UNIX days
  - SCCS: 1972, Bell Labs, Marc J. Rochkind
  - diff: 1974, AT&T, Hunt-McIlroy algorithm
  - RCS: 1982, GNU, Walter F. Tichy
  - patch: 1985, Larry Wall
  - CVS: 1986, Dick Grune
  - Subversion: 2000, CollabNet, Apache
  - Git: 2005, Linus Torvalds

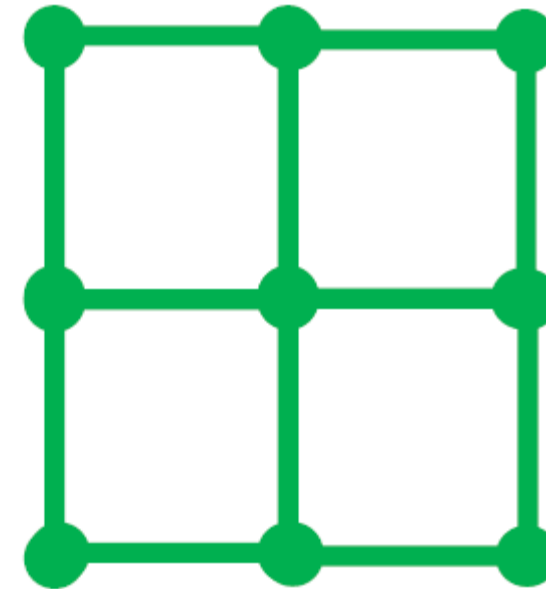
# Types of VCS



IT Learning &  
Outsourcing Center



Centralized Version  
Control Systems



Distributed Version  
Control Systems

[https://www.youtube.com/watch?v=\\_yQlKEq-Ueg](https://www.youtube.com/watch?v=_yQlKEq-Ueg)

# How does it work?



File Difference Settings Help

Compare Files Save Save All Previous File Next File Previous Difference Next Difference Unapply All Unapply Difference

Navigation

Source Folder	Destination Folder	Source File	Destination File	Source Line	Destination Line	Difference
/tmp/Development/	/tmp/Development/	old.txt	new.txt	1	1	Changed 3 lines
				11	14	Changed 1 line
				15	18	Deleted 4 lines
				22	21	Inserted 4 lines

old.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer di
2 ultrices. Pellentesque laoreet nunc in diam pharetra dignissim. Sus
3 elit. Nam nulla tortor, posuere sed ornare vel, tincidunt non ligul
4
5 Vivamus quis turpis tortor, non hendrerit felis. Nam dictum, libero
6 Maecenas tincidunt tempus pharetra. Donec purus nisl, tincidunt eu
7 Suspendisse a mollis urna. Curabitur scelerisque, sem sit amet comm
8 pretium sed faucibus in, vulputate nec nisl. Curabitur eleifend vel
9 et eros. Morbi tempus hendrerit ipsum, vel gravida sem accumsan acc
10
11 Sed adipiscing, nibh at ultrices porta, ipsum arcu dictum arcu, sit
12 vestibulum sem sit amet enim iaculis rutrum nec id nunc. Aliquam er
13 volutpat id ullamcorper nec, vehicula eu magna.
14
15 Quisque consequat varius risus, vitae condimentum sapien viverra ve
16 tempor. Nunc commodo suscipit elementum. Aenean lobortis nisl quis
17 ullamcorper mauris id nisl lacinia dapibus.
18
19 Nulla vehicula vestibulum elit vel dapibus. Fusce pulvinar nisl ac
20 Pellentesque nec justo ipsum, id euismod dui. Nam magna lectus, vol
21 porta. Mauris malesuada mi tempus nibh pharetra feugiat. Vestibulum
```

new.txt

```
1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque el
2 sagittis. Quisque blandit tempus luctus. Etiam ornare euismod arcu
3 hendrerit sem cursus enim malesuada ac fermentum eros aliquet. Cras
4 Nam porttitor, sem at adipiscing fermentum, nunc sem placerat torto
5 Aliquam tristique turpis ac purus convallis sed malesuada quam conv
6 dignissim egestas eu in orci.
7
8 Vivamus quis turpis tortor, non hendrerit felis. Nam dictum, libero
9 Maecenas tincidunt tempus pharetra. Donec purus nisl, tincidunt eu
10 Suspendisse a mollis urna. Curabitur scelerisque, sem sit amet comm
11 pretium sed faucibus in, vulputate nec nisl. Curabitur eleifend vel
12 et eros. Morbi tempus hendrerit ipsum, vel gravida sem accumsan acc
13
14 Sed and awk adipiscing, nibh at ultrices porta, ipsum arcu dictum a
15 vestibulum sem sit amet enim iaculis rutrum nec id nunc. Aliquam er
16 volutpat id ullamcorper nec, vehicula eu magna.
17
18 Nulla vehicula vestibulum elit vel dapibus. Fusce pulvinar nisl ac
19 Pellentesque nec justo ipsum, id euismod dui. Nam magna lectus, vol
20 porta. Mauris malesuada mi tempus nibh pharetra feugiat. Vestibulum
21
22 In pellentesque lacus sit amet dui dictum vulputate. Donec tempus m
23 tincidunt. Vestibulum ante ipsum primis in faucibus orci luctus et
24 eros eget mauris interdum ultrices. Ut eleifend pharetra sodales.
```

Comparing file file:///tmp/Development/old.txt with file file:///tmp/Development/new.txt

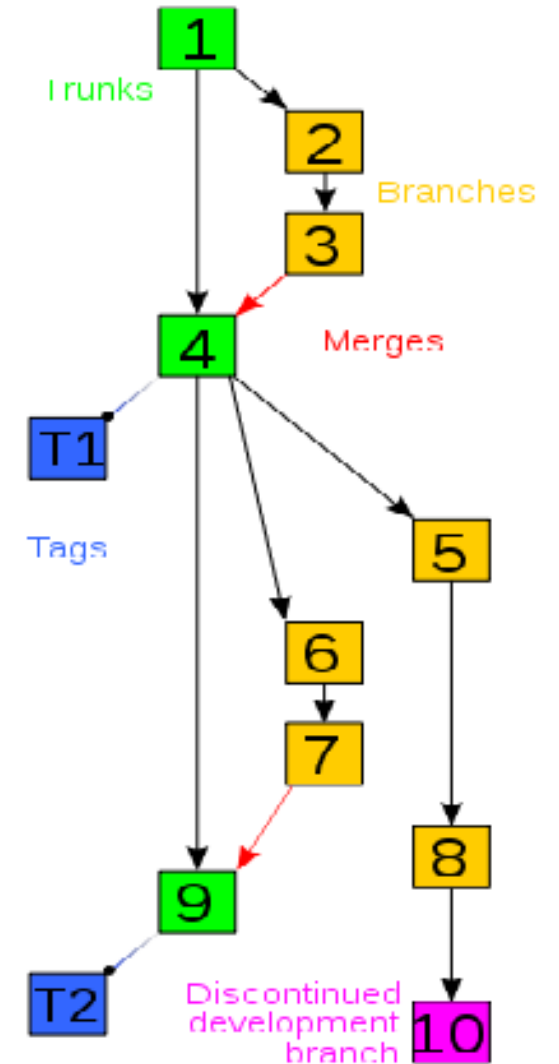
1 of 4 differences, 0 applied 1 of 1 file



# How does it work?



- Repository
- Revisions
- Tree baseline
- Branches
- Tags
- Wikipedia: [Version control](#)



# How does it work?



```
2011-08-21 08:55 Mark Story      o Adding a constant for TIME_START as $_SERVER[REQUEST_TIME] is only
2011-07-31 14:49 Mark Story      o Removing APP_PATH it is always the same as APP.
2011-07-28 13:44 Mark Story      o Re-adding support for PHP's include_path.
2011-07-26 01:46 Jose Lorenzo Rodr~ o Updating all @package annotations in doc blocks
2011-07-15 20:44 Jose Lorenzo Rodr~ o Fixing caching of class loading in App class, this was broken after a recent refactoring
2011-05-29 17:31 Juan Basso      o Updating the copyright to 2011.
2011-04-17 13:14 AD7six         o remove CAKE_TESTS constant
2011-04-17 13:13 AD7six         o remove CONFIGS constant
2011-04-17 12:37 AD7six         o correct greedy replace
2011-04-17 12:35 AD7six         o remove LIBS constant
2011-04-17 12:29 AD7six         o remove LIB_TESTS constant
2011-04-17 12:28 AD7six         o remove MODEL_TESTS
2011-04-17 12:27 AD7six         o remove HELPER_TESTS constant
2011-04-17 12:27 AD7six         o remove COMPONENT_TESTS constant
2011-04-17 12:26 AD7six         o remove CONTROLLER_TESTS constant
2011-04-17 12:25 AD7six         o remove CAKE_TEST_LIB constant
2011-04-17 12:18 AD7six         o remove CONSOLE_LIBS constnat
2011-04-17 12:17 AD7six         o remove ELEMENTS constant
2011-04-17 12:10 AD7six         o remove LAYOUTS constant
2011-04-17 12:10 AD7six         o remove HELPERS constant
2011-04-17 12:09 AD7six         o remove VIEWS constant
2011-04-17 12:06 AD7six         o remove BEHAVIORS contant
2011-04-17 12:05 AD7six         o remove MODELS constant
2011-04-17 12:04 AD7six         o remove COMPONENTS constant
2011-04-17 11:13 AD7six         o remove CONTROLLERS constant
2011-05-13 02:53 Jose Lorenzo Rodr~ o Search and replace for changed paths
2011-05-13 01:53 Jose Lorenzo Rodr~ o Starting unification of casing in remaining folders
2011-04-11 22:15 Jose Lorenzo Rodr~ o Fixing some bake tasks
2011-04-10 15:32 Jose Lorenzo Rodr~ o Renaming constants and fixing the last error related to folder renaming
2011-03-09 22:11 Jose Lorenzo Rodr~ o Properly testing the App::build() method and bugfixing some issues
2011-01-02 02:00 José Lorenzo Rod~ o Merge remote branch 'origin/2.0' into 2.0-class-loading
2010-12-11 01:17 José Lorenzo Rod~ o Fixing several errors when running the testsuite
2010-12-07 01:26 José Lorenzo Rod~ o Making shells run again
2010-12-05 10:54 José Lorenzo Rod~ o Starting to change the class loading for app classes
2010-12-04 02:34 José Lorenzo Rod~ o More replacements to get class loaded using the new file scheme
2010-12-04 02:16 José Lorenzo Rod~ o Moving more classes to the new layout, it is time to bring the debugger up
[main] 0cb70ae3bb22c40fc69c8882c1d48657cce5bed - commit 8 of 39 (92%)
```

# How does it work?



```
2008-05-30 11:40 nate 42da5ab 1 <?php
2008-05-30 11:40 nate 42da5ab /**
2008-05-30 11:40 nate 42da5ab  * Basic Cake functionality.
2008-05-30 11:40 nate 42da5ab  *
2009-05-01 14:05 gwoo 15518b8 5  * Handles loading of core files needed on every request
2008-05-30 11:40 nate 42da5ab  *
2010-10-03 12:38 mark_story f847080  * PHP 5
2008-05-30 11:40 nate 42da5ab  *
2009-05-01 14:05 gwoo 15518b8  * CakePHP(tm) : Rapid Development Framework (http://cakephp.org)
2011-05-29 17:31 Juan Basso 192812e 10  * Copyright 2005-2011, Cake Software Foundation, Inc. (http://cakefoundation.org)
2008-05-30 11:40 nate 42da5ab  *
2008-05-30 11:40 nate 42da5ab  * Licensed under The MIT License
2008-05-30 11:40 nate 42da5ab  * Redistributions of files must retain the above copyright notice.
2008-05-30 11:40 nate 42da5ab  *
2011-05-29 17:31 Juan Basso 192812e 15  * @copyright      Copyright 2005-2011, Cake Software Foundation, Inc. (http://cakefoundation.org)
2010-01-26 17:03 Mark Story cdd8784  * @link          http://cakephp.org CakePHP(tm) Project
2011-07-26 01:46 Jose Lorenzo Rodr~ cfd2d9e  * @package       Cake
2008-10-30 17:30 AD7six 3f2fa69  * @since         CakePHP(tm) v 0.2.9
2009-05-01 14:05 gwoo 15518b8  * @license       MIT License (http://www.opensource.org/licenses/mit-license.php)
2008-05-30 11:40 nate 42da5ab 20  */
2011-08-21 08:55 Mark Story 8503ffd define('TIME_START', microtime(true));
2011-08-21 08:55 Mark Story 8503ffd
2009-07-31 11:48 nate 729d8fd if (!defined('E_DEPRECATED')) {
2009-08-01 08:36 nate 7847044     define('E_DEPRECATED', 8192);
2009-07-31 11:48 nate 729d8fd 25 }
2009-08-01 08:36 nate 7847044 error_reporting(E_ALL & ~E_DEPRECATED);
2009-08-01 08:36 nate 7847044
2011-07-28 13:44 Mark Story 721c438 if (!defined('CAKE_CORE_INCLUDE_PATH')) {
2011-07-28 13:44 Mark Story 721c438     define('CAKE_CORE_INCLUDE_PATH', dirname(dirname(__FILE__)));
2011-07-28 13:44 Mark Story 721c438 30 }
2011-07-28 13:44 Mark Story 721c438
2011-07-28 13:44 Mark Story 721c438 if (!defined('CORE_PATH')) {
2011-07-28 13:44 Mark Story 721c438     define('CORE_PATH', CAKE_CORE_INCLUDE_PATH . DS);
2011-07-28 13:44 Mark Story 721c438 }
2011-07-28 13:44 Mark Story 721c438 35
2011-07-28 13:44 Mark Story 721c438 if (!defined('WEBROOT_DIR')) {
[blame] lib/Cake/bootstrap.php - line 24 of 154 (23%)
```

# Which one should I use?



- Git
- If you can't use Git, use anything else
- Make sure you do use something!
- Git is free and Open Source
- Git is distributed
- Git is cross-platform (Linux, Windows, ...)
- Git is very fast. And smart.
- Git is de fact standard in Open Source world
- Git is on the GitHub

# How do I start using it?



- You are probably already using it via:
  - Microsoft Office, OpenOffice.org, KOffice, WordPress, Drupal, Joomla, or Wiki engine.
- Install Git
- Tutorials, book (<https://book.git-scm.com/book/en/v2>)
- Try and play
- Register GitHub account
- Enjoy!

# How do I start using it?



- Once Git is installed, use Git Bash on your computer and type the following commands for initial configuration:

```
$ git config --global user.name "Milen Strahinski"  
$ git config --global user.email "milen.strahinski@pragmatic.bg"
```

# Most used commands. Lets see them...



IT Learning &  
Outsourcing Center

\$ git init - makes the directory in which you write that command to be a git tracked project

\$ git status - shows you if you have untracked or uncommitted changes at any point in time

\$ git add SOME\_FILE - makes a specific file tracked(staged) by git

\$ git add . - makes all the files in the current directory being tracked(staged) by git

\$ git commit -m "some comment" - updates any changes you've done to some file in the local repo

\$ git remote add origin <http://some-git-repo.com/projectname.git> - add the remote repository which will be used to store(upload/push) the project(make sure this repo is already existing)

\$ git push origin master - uploads/pushes all of the added files from the local repository into the remote repository. Word „origin“ is an alias of the remote repo you're working with

\$ git clone <http://gitrepo/gitproject.git> - clones an existing project from a remote place into your local computer and you can start working on it together with your team.

\$ git pull origin master - this will automatically fetch any new remote branches from your colleagues and they will become ready for checkout and additionally will download/update/pull all the changes that your colleagues have pushed in the specific branch you're currently in.

# How to ignore some files



- Often you want some of the files not to be staged/tracked by Git and they should never get into the remote repository. In order to achieve that you need to create a file with name

`.gitignore` - including the `.` at the beginning of the filename (if Windows does not allow you to create that file, use Git Bash to create the file by typing "touch .gitignore", then you can edit it freely)

and put in that file something like

`*.class`  
`bin/`

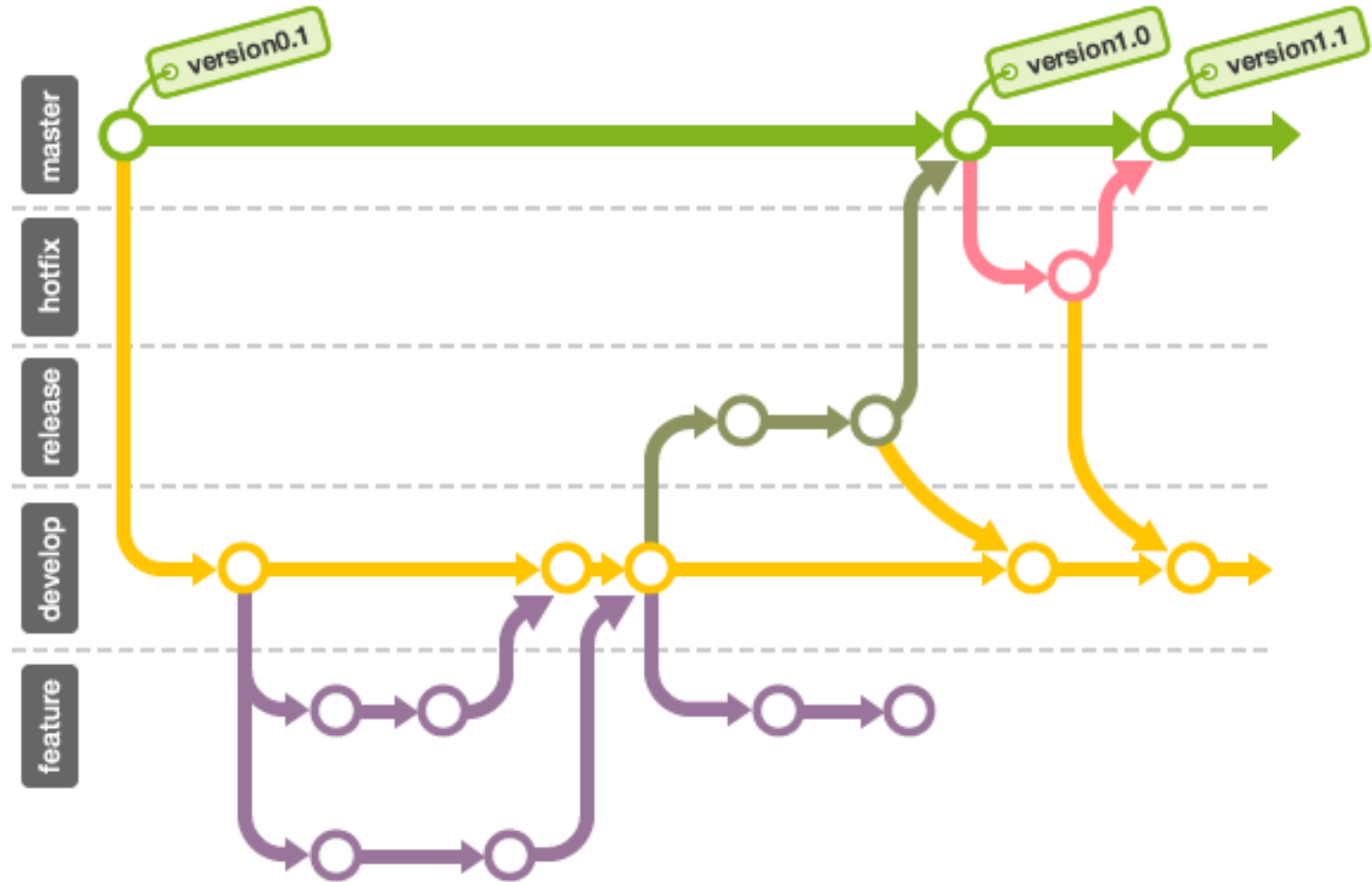
that way it will automatically ignore all the compiled files not to be pushed into the remote repository.



# What is a branch?



- Branching means you diverge from the main line of development and continue to do work without messing with that main line. A branch in Git is simply a lightweight movable pointer(reference) to one of these commits. The default branch name in Git is master. As you start making commits, you're given a master branch that points to the last commit you made. Every time you commit, it moves forward automatically.



# Creating, switching, updating branches?



- Let's say we're currently in the master branch, and we want to create a new one out of it. Simply write:

```
$ git branch some_branch_name - this will create a new branch out of the current state of the master you're currently in
```

- Checking what branches are available locally:

```
$ git branch -a - this will list all the available branches you can switch to
```

- Switching from one branch to another

```
$ git checkout branch_name - once you checked the available branches you can switch to them using that specific command
```

- If someone has pushed a code and you want to update your project and receive the latest changes from your colleagues in the specific branch you are in, just type:

```
$ git pull - this will automatically fetch any new remote branches from your colleagues and they will become ready for checkout and additionally will download/update/pull all the changes that your colleagues have pushed in the specific branch you're currently in.
```

# Merging branches



- Before you merge make sure you're in the destination branch, or in other words if you want to get the changes from branch1 and have them into master, you need to checkout the master first. Then:

```
$ git merge name_of_source_branch - this will take all the changes you've done in  
name_of_source_branch and merge them into master
```

```
$ git commit -am „comment“- if there is a CONFLICT during the merge, when you  
resolve it you need to commit first
```

```
$ git push - after every merge make sure to push
```

# How do I start using it?



- One lovely explanation on how to use Git: [https://www.youtube.com/watch?v=YgXZQO1n\\_7c](https://www.youtube.com/watch?v=YgXZQO1n_7c)
- In order to start a completely new project you need to initialize your directory to be a Git tracked project:

```
$ mkdir Project_Name - creating the directory in which your project lives
```

```
$ cd Project_Name - enter into that specific directory
```

```
$ git init - initialize the dir as a git project that you can start tracking
```

- In order to make an existing project on your local computer to be part of a Git repository:

```
$ cd Your_Project_Directory - enter into that existing project directory
```

```
$ git init - initialize the dir as a git project that you can start tracking
```

```
$ git remote add origin http://some-git-repo.com/projectname.git - add the remote repository  
which will be used to store(upload/push) the project(make sure this repo is already  
existing)
```

```
$ git add . - adds all the files in the current directory to start tracking them
```

```
$ git commit -m "some commit message for the initial commit" - commits the files in the  
local repository with a specific message on what about the commit is
```

```
$ git push origin master - uploads/pushes all of the added files from the local repository  
into the remote repository
```

# How do I start in IntelliJ? (part 1)



- Great video which makes an overview on how to work with IntelliJ when using Git:

<https://www.youtube.com/watch?v=uUzRMOCBorg>

# Merge branches with conflicts?



- Sometimes you're merging branches and there are changes on the same line in both branches and as a result it conflicts, luckily IntelliJ has an integrated UI that guides us through the conflict resolution:

<https://www.youtube.com/watch?v=opR6V4G2aJM>

# How to work with GitHub? Let's check it?

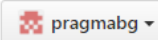


IT Learning & Outsourcing Center

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

myfirstproject

Great repository names are short and memorable. Need inspiration? How about **automatic-waddle**.

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

The screenshot displays the GitHub interface. At the top, there's a navigation bar with 'This organization', a search bar, and links for 'Pull requests', 'Issues', and 'Gist'. Below this, the 'postplanner' organization page is shown, featuring a 'News Feed' with a list of recent activity, including branch creations and deletions by users like n1kolai and yashamuru. On the right, a 'Repositories' sidebar lists 'postplanner/staging', 'postplanner/postplanner\_scheduler', and 'postplanner/hiring-project'. The main content area shows a recent push to the 'develop' branch by 'yashamuru'. Below the activity feed, the 'Quick setup' section provides instructions for cloning the repository, including a 'Set up in Desktop' button, an SSH key setup link, and a terminal command: `git clone https://github.com/pragmabg/myfirstproject.git`. It also offers a 'New repository' button and a 'Create new repository' button. The bottom section, '...or push an existing repository from the command line', provides a terminal command: `git remote add origin https://github.com/pragmabg/myfirstproject.git` and a 'Push' button. The final section, '...or import code from another repository', mentions Subversion, Mercurial, or TFS projects and includes an 'Import code' button.

# How to delete GitHub repo?



- Copy the repository URL from GitHub. It looks something like:  
<https://github.com/pragmabg/myfirstgitproject.git>
- Then open in the browser the following link based on the URL above:  
<https://github.com/pragmabg/myfirstgitproject/settings>
- Scroll down to the bottom to the “Danger Zone” section and click on the “Delete This Repository” button and confirm the name of the repository by typing it in the field.
- For more information refer to the official GitHub documentation on how to delete a repository:
- <https://help.github.com/articles/deleting-a-repository/>



# Questions?



IT Learning &  
Outsourcing Center



# Course Feedback



IT Learning &  
Outsourcing Center

- Please, share your overall feedback at the bottom of the course page at:

<http://pragmatic.bg/courses/automated-testing-course/>

HIGHLY APPRECIATED! 😊



# Course Certificates



IT Learning &  
Outsourcing Center



😊 Congratulations! 😊