

# Computer Programming II

Version control and git

Sven-Erik Ekström

Tom Smedsaas

Uppsala universitet

# Version control, why?

There are many reasons to use version control when you develop code,

- Backup
- History, how has the code evolved
- Something worked before, but stopped working, what has changed?
- Cooperation, several people or organizations work with the same code at the same time
- Who did a particular change in the code, and when?
- Work on several things at the same time (branches)
- Release code in different versions to clients.

# Version control, SCM

There has been many different systems for version control (**S**ource **C**ode **M**anagement), for example,

- CVS, [https://en.wikipedia.org/wiki/Concurrent\\_versions\\_system](https://en.wikipedia.org/wiki/Concurrent_versions_system)
- SVN, [https://en.wikipedia.org/wiki/Apache\\_Subversion](https://en.wikipedia.org/wiki/Apache_Subversion)
- Hg (Mercurial), <https://en.wikipedia.org/wiki/Mercurial>
- git, <https://en.wikipedia.org/wiki/Git>

All these (and others) are still used in industry, but git is the most popular.

Reason to use it in MA4 is for you to try it, and hopefully you can start using it yourself in coding projects.

# Version control, Repository `git`

A repository is a “directory” where you keep code for a specific project (e.g., one course or one application you develop). This directory will keep a full history, with snapshots, of the code’s evolution.

One can either create repositories locally on your computer, or you can use providers of server hosted repositories. Also, some companies have their own servers, to protect data.

Then, you typically synchronize the code between repositories.

# Version control, Account git

Some popular providers of git servers are,

- <https://bitbucket.org/>
- <https://gitlab.com/>
- <https://github.com/>

Get an account at some provider (if you already have an account, then use that). Github is recommended <https://education.github.com/pack>. Use a private email so you can keep the account after your studies. Choose a “good” username.

# Version control, client `git`

You need a client (program) to be able to use `git`. There are many different alternatives and you choose yourselves what you use, here are some suggestions.

	macOS	linux	windows	
<code>git</code> (CLI)	x	x	x	not graphical
VSCode	x	x	x	Recommended
Smartgit	x	x	x	
GitKraken	x	x	x	
Sourcetree	x		x	
Github Desktop	x		x	

<https://git-scm.com/downloads/guis/>

# Version control, client `git` with VSCode

To use `git` from within VSCode, you need `git` installed on your machine.

Linux and macOS typically has it installed by default. If not on macOS, then we recommend that you install it thru Homebrew (<https://brew.sh/>) or XCode thru the App Store.

On Windows we recommend the official installation from <https://git-scm.com/>.

Once installed on your machine VSCode will handle `git` automatically (may ask to install a `git` extension, then just accept).

# Version control, Repository git

Some popular **open** repositories are

- <https://www.openhub.net/p/git>
- <https://github.com/microsoft/vscode>
- <https://github.com/torvalds/linux>



# Version control, Repository git

Some popular **open** repositories are

- <https://www.openhub.net/p/git>
- <https://github.com/microsoft/vscode>
- <https://github.com/torvalds/linux>

You will create a **private** repository for this course/assignment.

Example: <https://github.com/sverek/prog2>

# Version control, Workflow

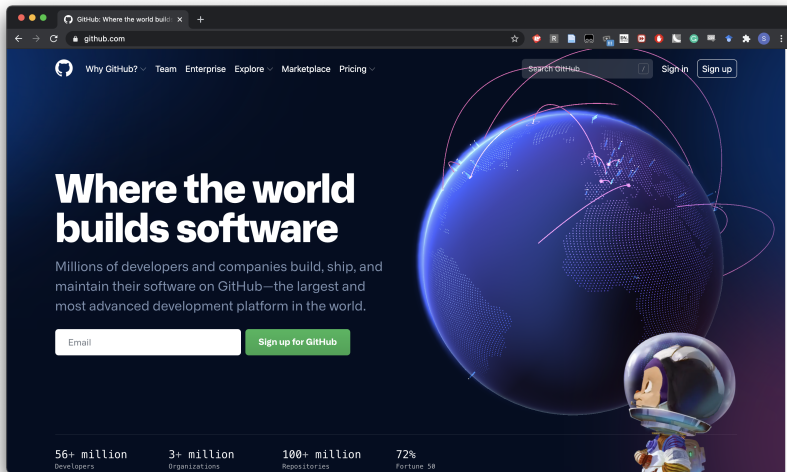
Demonstration of workflows.

# Version control, More information git

Some links where you learn more

- <https://hackernoon.com/understanding-git-fcffd87c15a3>
- <https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>
- <https://git-scm.com/docs/gittutorial>

# Version control, Github [github.com](https://github.com)



# Version control, Github [github.com](https://github.com)

