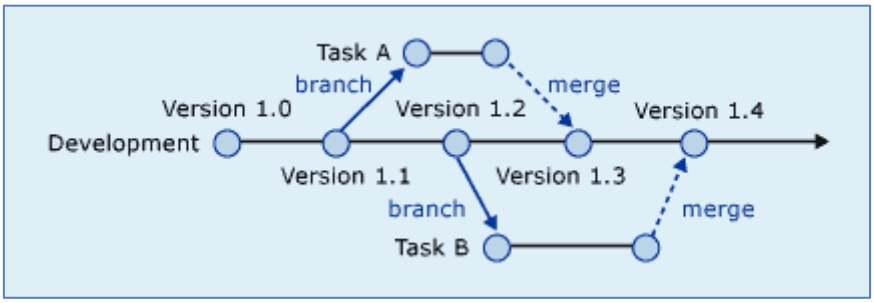
GitHub

We are using the “Branch per Task” concept. That means that every development task is a new, independent branch. Tasks are merged into the permanent master branch as they are completed.



Clone the project to your hard drive

This is done just once in the beginning.

1	`git clone <clone_url>`	Clone the project to your hard drive

clone_url = `https://github.com/Fille881/Flying-Dutchman-4.git`

Start working on a new branch

Every time you start working on a new task, you create your own independent branch of the master branch. This means that you never work directly on the master branch.

1	`git branch <branch_name>`	Create a new branch (The name should begin with your name) e.g. *Tom-outOfStockUI* or just *Tom*
2	`git checkout <branch_name>`	Switch to the created branch

Alternative:

1	`git checkout -b <branch_name>`	Create a new branch and switch to the created branch

Save changes in your branch

While you are working on a task you save your changes in the files, but from time to time you also have to save them in your own branch.

1	`git add .`	Prepare all files for commit
	`git add <filename>`	Prepare a specific file for commit
2	`git commit -m "<comment>"`	Commit the changes in your own branch (with a comment about what you changed)

Merge your branch with the master branch

When you finished your task, you merge your branch with the master branch and delete your own branch. If no merge conflicts appear, that's fine. Otherwise you have to solve the conflicts.

1	`git checkout master`	switch to master branch
2	`git pull`	get the latest code of the master branch
3	`git merge <branch_name>`	merge your changes into the master branch
4	`git push`	push the merge-result in the master branch
5	`git branch -d <branch_name>`	delete your branch (only if it was merged successfully)

Solve merge conflicts

That's the only time you are working directly on the master branch.
Don't try that at home, if you are not 100% sure what you are doing. :)

1		Find and solve the conflicts in the files
2	`git add .`	Prepare all files for commit
	`git add <filename>`	Prepare a specific file for commit
3	`git commit -m "<comment>"`	Commit the changes in the master branch (with a comment about what you changed)
4	`git push`	push the solved merge conflicts in the master branch