

# Advantages of Version Control Systems

useful also for non-hardcore programmers

`stephan.gabler@gmail.com`

December 6, 2010

# Why using Version Control

## Typical Problems

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote it
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## Typical Problems

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote it
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## Typical Problems

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote it
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## Typical Problems

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote it
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

So, what is it ?

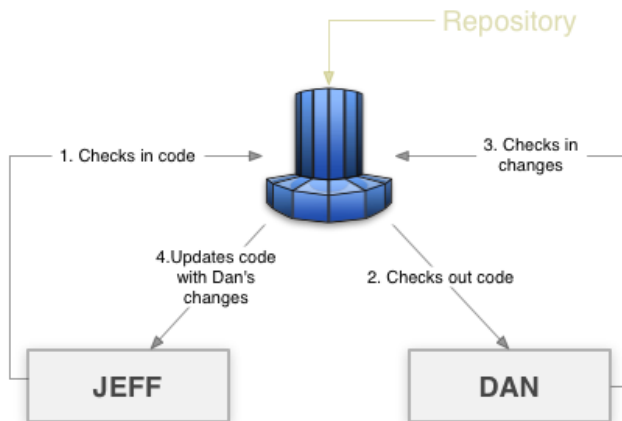
## Git

A central place for code, and **its history** !

You can work on the same files together, without merging on the next day, you can track your changes, changes of others and can go back to whatever previous version you want. And you have a central place for keeping all your code  
blackboard diagram

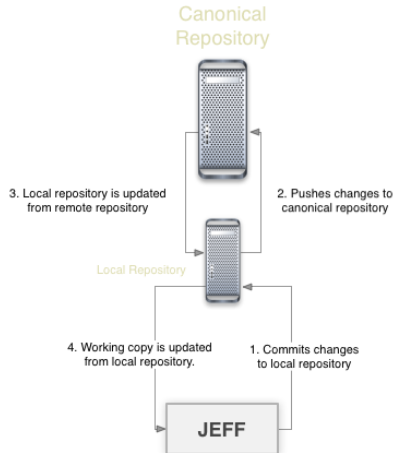
# Version Control Workflow

## Basic



# Version Control Workflow

## Distributed





# Why using Version Control

## How it helps

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote that line
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## How it helps

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote that line
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## How it helps

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote that line
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Why using Version Control

## How it helps

- 1 Where do I get the latest (and running) version of a program
- 2 How can we work together on some code, without always sending files, ect..
- 3 I don't understand this code, who wrote that line
- 4 S\*\*\*, I made some *improvements* and now it does not work anymore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore



# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Some Guidelines

that make it easier for everybody

- commit often
- pull often
- only running code
- code vs. results
- comments
- commit messages
- gitignore

# Remarks

- can be used also with binary data (images, poster, data)
- also local repositories are really useful
- the homework will help you to get some experience
- play with it, it can really help you and the whole lab a lot
- ask me whenever you have questions

# Remarks

- can be used also with binary data (images, poster, data)
- also local repositories are really useful
- the homework will help you to get some experience
- play with it, it can really help you and the whole lab a lot
- ask me whenever you have questions

# Remarks

- can be used also with binary data (images, poster, data)
- also local repositories are really useful
- the homework will help you to get some experience
- play with it, it can really help you and the whole lab a lot
- ask me whenever you have questions

# Remarks

- can be used also with binary data (images, poster, data)
- also local repositories are really useful
- the homework will help you to get some experience
- play with it, it can really help you and the whole lab a lot
- ask me whenever you have questions

# Remarks

- can be used also with binary data (images, poster, data)
- also local repositories are really useful
- the homework will help you to get some experience
- play with it, it can really help you and the whole lab a lot
- ask me whenever you have questions