Institute of Acoustics and Speech Communication
Chair of Speech Technology and Cognitive Systems

Introduction
Day 1

Simon Stone

7/19/2021

Agenda
Day 1

Introduction or: What‘s the point of all this?

Test-driven development

Version control using Git and GitHub

Mapping the problem

Pair programming phase

Introduction or:
What‘s the point of all this?

Introduction or:
What‘s the point of all this?

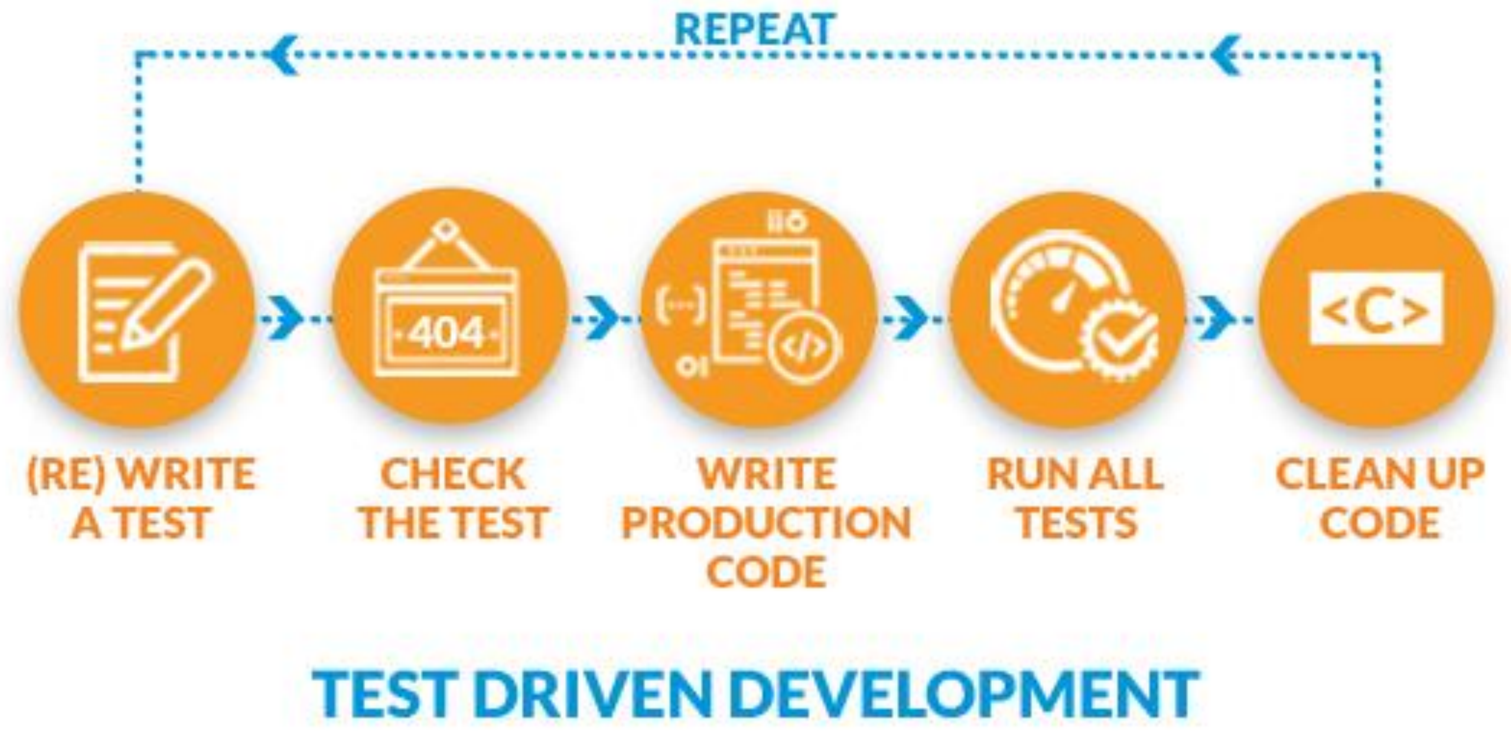
- **Write better code!**
- Learn to think more in terms of interfaces and interactions than building monolithic „one-trick-ponies“
- Learn professional tools of the trade and concepts
- Create a useful addition to the Chair‘s toolbox along the way

Test-driven development

Test-driven development

Basic idea:

- **First comes the test, then comes the code!**



Source: https://www.linkedin.com/pulse/test-driven-development-tdd-why-you-should-care-lance-harvie

Test-driven development

How do you „write a test“?

Unit testing:

- You programs should consist of individual „**units**“ (think: building blocks in a flow chart) that can be individually tested using some given input and some expected output.
- Some languages have support for this already built-in
- In C++, various frameworks exist for this purpose.
- The frameworks usually consist of a bunch of objects and macros to quickly generate a test program.
- We will be using Google Test for this project.

Version control using Git and GitHub

Version control using Git and GitHub

Git:

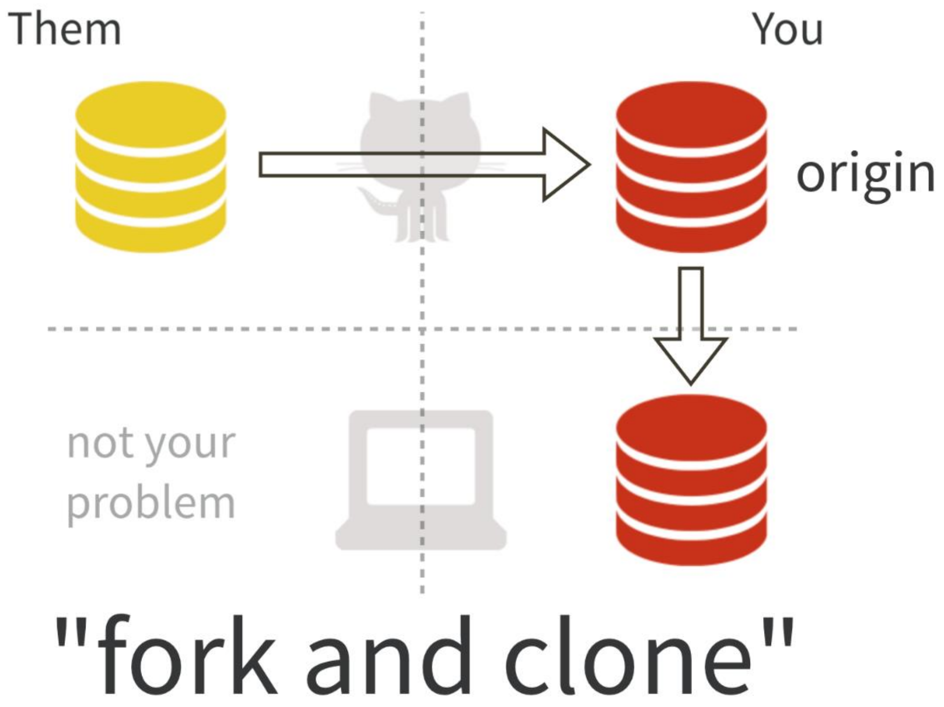
A protocol and command line tool to track changes in a software project involving many devs.

GitHub:

An online platform hosting software projects, documentation, and much more using the git protocol.

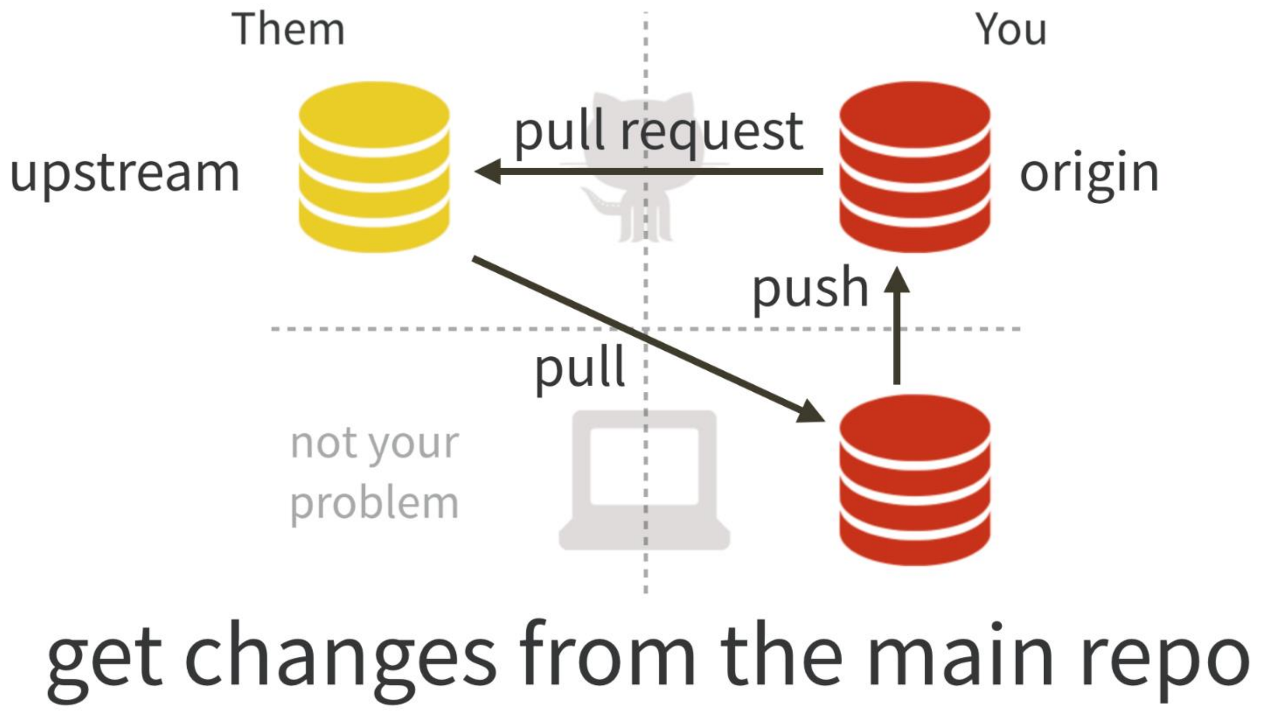
Version control using Git and GitHub

How to git?



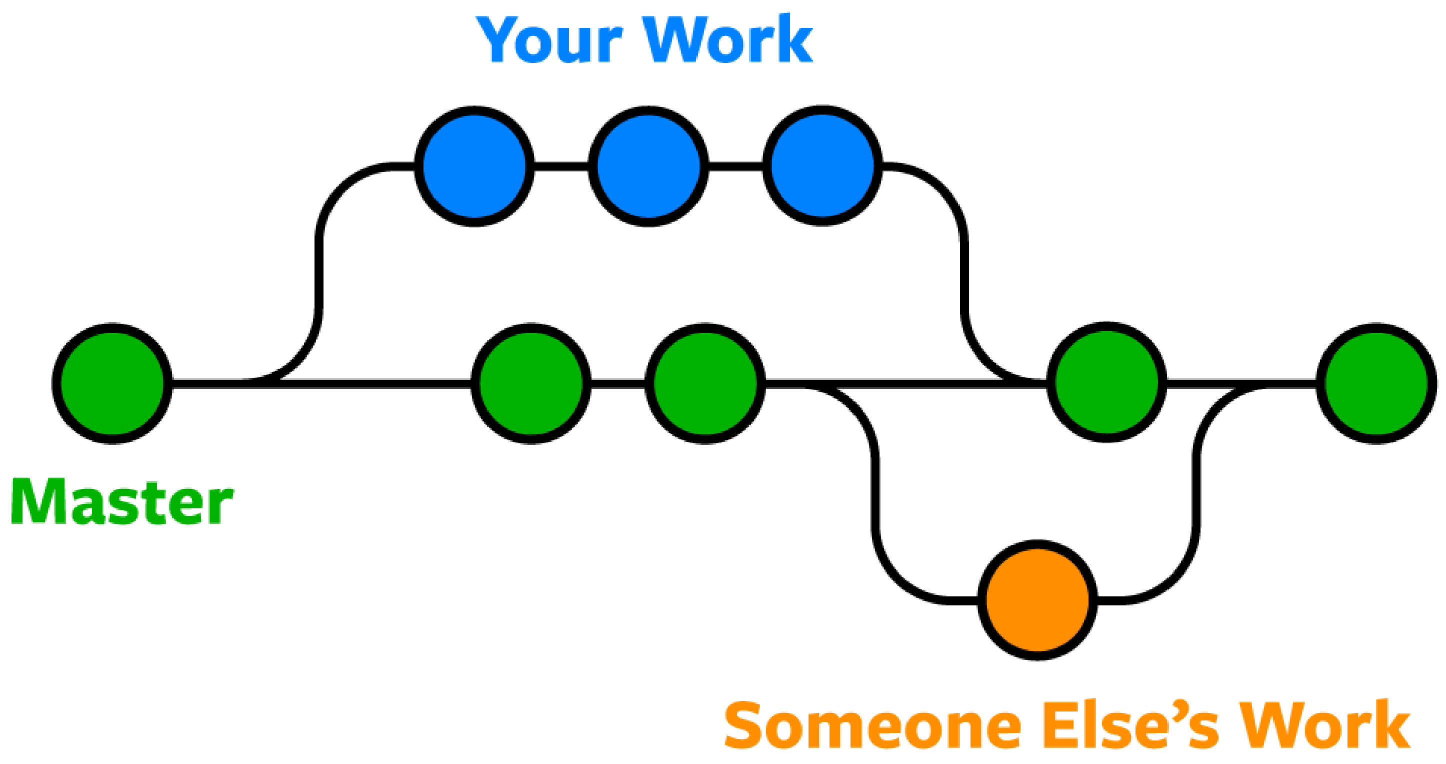
Version control using Git and GitHub

How to git?



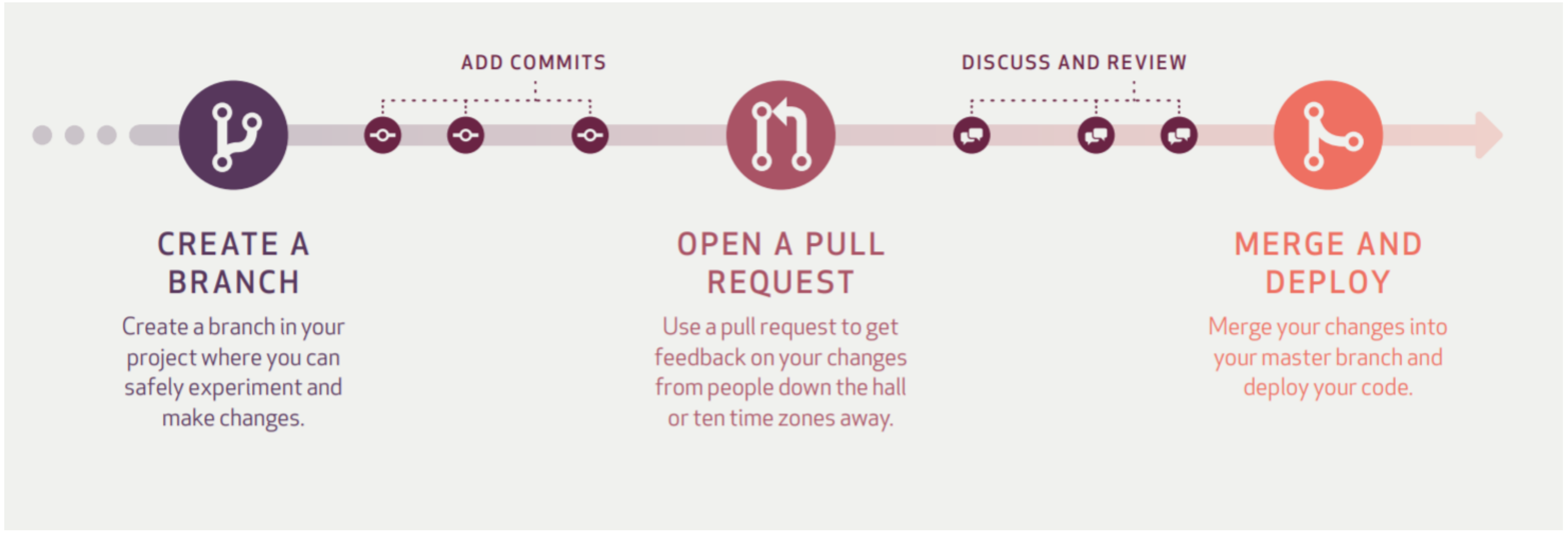
Version control using Git and GitHub

How to git?



Version control using Git and GitHub

How to git?


ADD COMMITS
DISCUSS AND REVIEW
CREATE A BRANCH
Create a branch in your project where you can safely experiment and make changes.
OPEN A PULL REQUEST
Use a pull request to get feedback on your changes from people down the hall or ten time zones away.
MERGE AND DEPLOY
Merge your changes into your master branch and deploy your code.

Mapping the problem

Mapping the problem

Step 0 for any programming problem:

- Think about how to **structure** your code!

Use mind maps, UML, pen and paper, your glorious mind palace, red string on a cork board, whatever.

But do it.

So let‘s do it:

https://drive.google.com/file/d/1zwpder3_g5eWmKweGtHMBdOyIvx8tfuX/view?usp=sharing

Pair programming phase

Pair programming phase

- Buddy up in teams of two
- Ideally one more experienced and one less experienced dev
- Move to a BigBlueButton breakout room
- Connect using Visual Studio Live Share or regular screen sharing
- One team member is the **driver**, who does the coding, and the other one is the **navigator**, who reviews the code as it is being produced.
- Switch roles regularly (using version control to keep your code synced)

Have fun!