Part 1

Think about how these networks are different. Analyze the "dimensions" of these networks. What are the relevant attributes (e.g., commits, users, branches, commit size, etc.) of these representations? What other attributes could be relevant in this graph? Write a list of all the attributes your visualization could show.

The top line represent the 'master' branch, and every side branch is represented by lower levels of lines. The difference in colours of those lines, represent the different users working in the repository. The end of lines containing an arrow are mergers between branches. The dots in each line represent pushes into those branches.

List of attributes that could be useful in this graph:

- Scrollbar to check in which position you are in the graph
- Allow users to select a scale in line with parameters (such as pushes, branches etc.) This makes it easier to overview important data you need
- An option to go to a certain year
- Ability to mark a push so that developers can easily see which push is more important (e.g. new stable release, new bèta release)

Are there different roles, i.e., different types of users who might want to achieve different things? Write a list of user roles.

Yes there are different roles, these are simulated by the owners on the left hand side.

List of user roles:

- Head developer
- Front-end developer
- Back-end developer
- Visualization developer

Think about which tasks a user of your visualization might want to achieve. Write down a list of tasks.

To get an overview of the overall developing stages based on your own preferences of interest. For example, as a back-end developer you would be less interested to see what the visualization developer is doing.

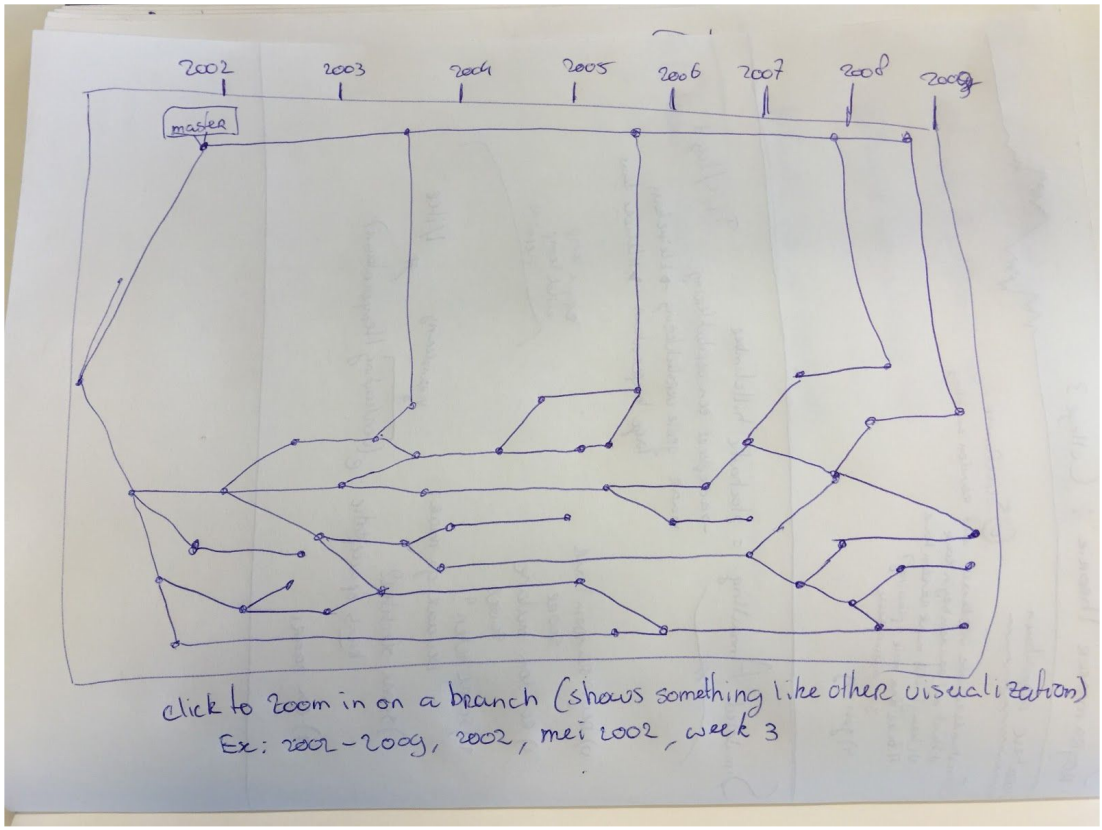
List of tasks:

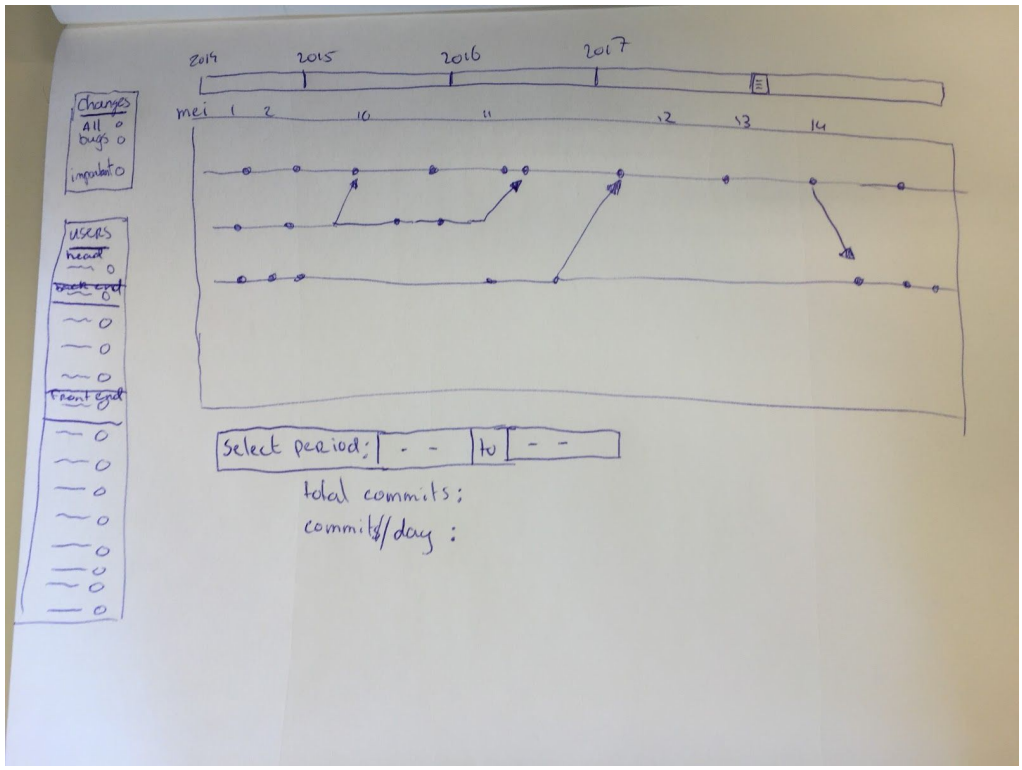
- An option to select a development department to only see what they are doing
- An option to see who is still active in the development program

Identify one role that you want to design your visualization for. Prioritize your task and attribute lists based on this role's needs.

The ability for the user to be more selective on the data in the graph. This could be done by data selection per active user, year, marked pushes.

Part 2


2002
2003
2004
2005
2006
2007
2008
2009
master
click to zoom in on a branch (shows something like other visualization)
Ex: 2002-2009, 2002, mei 2002, week 3


2014
2015
2016
2017
mei
1
2
10
11
12
13
14
Changes
All 0
bugs 0
important 0
USERS
heart
0
Back end
0
0
0
0
Front end
0
0
0
0
0
0
0
0
0
Select period; - - to - -
total commits:
commit#/day :