

## Data Processing: Design 3

### Part 1: Analysis

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.

On the left side is a list of users in the repository. For each of the users a graph of commits is drawn. Every point in the graph represents a single commit. The difference in the two graphs lies in how the members of the repository update their project. In the first graph, different team members work on different branches, which can be seen by the many different colored lines, while in the second graph there is only one line for the most part.

Relevant attributes in the visualizations are: commits, users, forks, branches, dates, branch/repository names and branch merges.

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

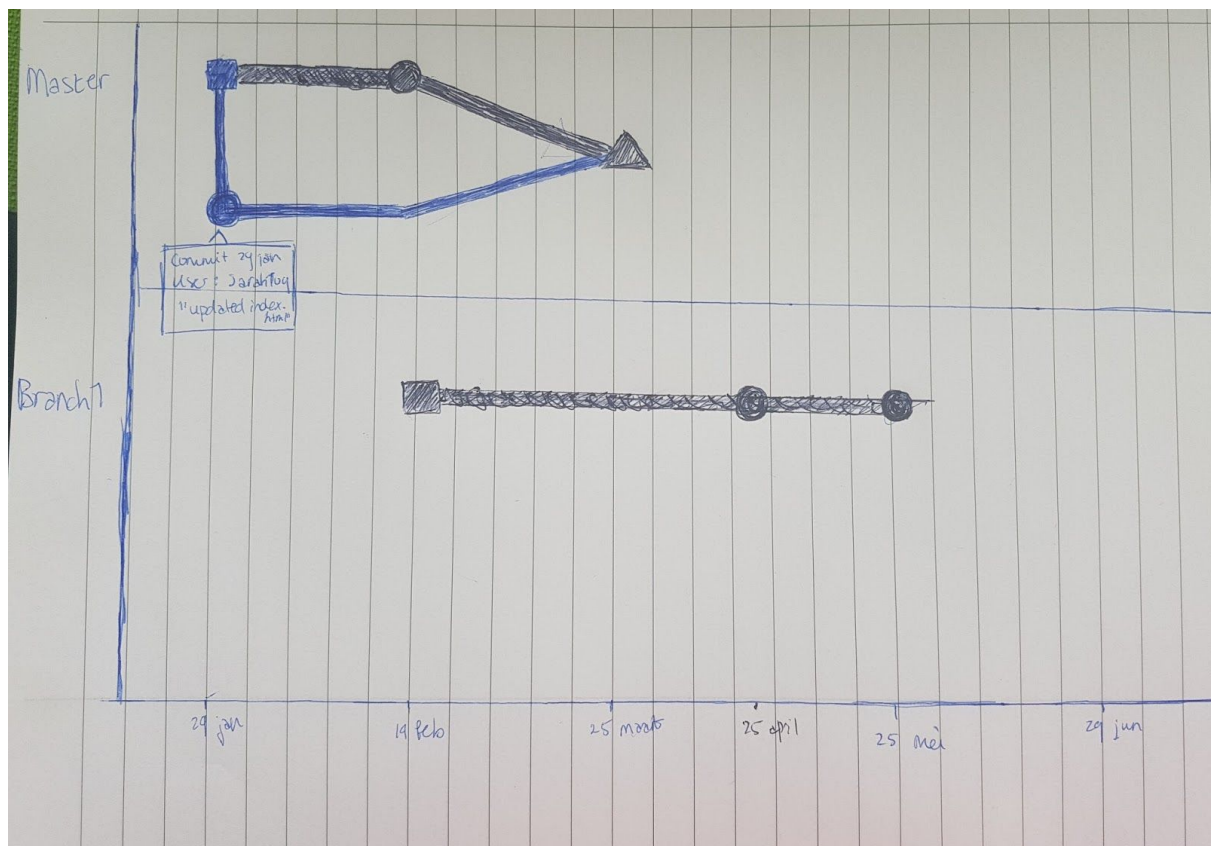
There might be a user who has more of a monitoring/supervising role (employer/manager/teacher) and there might be a user who has more of a executing/performing role (employee/student).

3. Think about which tasks a user of your visualization might want to achieve. Write down a list of tasks.
  - Get an overview of the changes in the repository
  - See who committed certain changes and/or who commits a lot of changes
  - See the history of the creation of the repository
  - See different merges and/or see who's changes merged with who's
  - See when new new branches or forks are created
  - View which members are part of the network and what branches they have
4. Identify one role that you want to design your visualization for. Prioritize your task and attribute lists based on this role's needs.
  - See who committed certain changes and/or commits a lot of changes

For example: a teacher who wants to see whether the students distributed the assignment mutually. Or a manager who wants to see whether the col-laboration between his employees runs smoothly.

## Part 2: Sketching

- The y-axis separates the different branches
- The x-axis shows the dates
- The square shows the creation of a branch
- The circle shows a commit
- Clicking on a circle/point shows which user committed the change and the commit message
- Different colored lines represent different users
- Two lines merging into one shows when different user works are merged, this is represented by a triangle
- Clicking on the triangle will show what changes were merged and by who



## Part 3: Group reflection

First of all, we thought it was important to have the x-axis show the dates, and the most important attribute we wanted to show was when commits are made. The different symbols (circle, square, triangle) show clearly what kind of change was made in the repository. Another attribute we thought was important was showing the different branches. In our visualization we chose to display the different branches separately with the names shown on the left side of the graph. Different users are marked out by colour and hovering over the symbols (circles, squares, triangles) provides more information about the commit/merge/new branch.