# PART 1 - ANALYSIS

- 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.
    - Branches
        - Branch name
    - Commits
        - Comments
        - User
    - Merges
        - Comments
        - User
    - Forks
        - Project name
    - Date
    - Projects

- Are there different roles, i.e., different types of users who might want to achieve different things? Write a list of user roles.
    - Master
        -
    - Developer
        - Inside the project
        - Outside the project
    - Tester
    - Designer
    - End user

- Think about which tasks a user of your visualization might want to achieve. Write down a list of tasks.
  - See where a branch or fork originates
  - When the developers have started writing code
  - How often something is committed
  - Is the code up to date
  - What kind of changes have been made
  - Who is working on what?
  - When something is committed.
  - What has been committed?
  - Who committed it?
  - Check the relevancy of the commits.
  - See how many forks or branches there are
  - See how popular the project is
- Identify one role that you want to design your visualization for. Prioritize your task and attribute lists based on this role's needs.
  - For the developers who forked the original project
    - Changes in the original project since forking
      - Amount of commits
      - Size of commits
      - Comments of commits
    - When is something forked
   - Date
      - How many days/weeks/years ago(?)

PART 2 - SKETCHING

1.
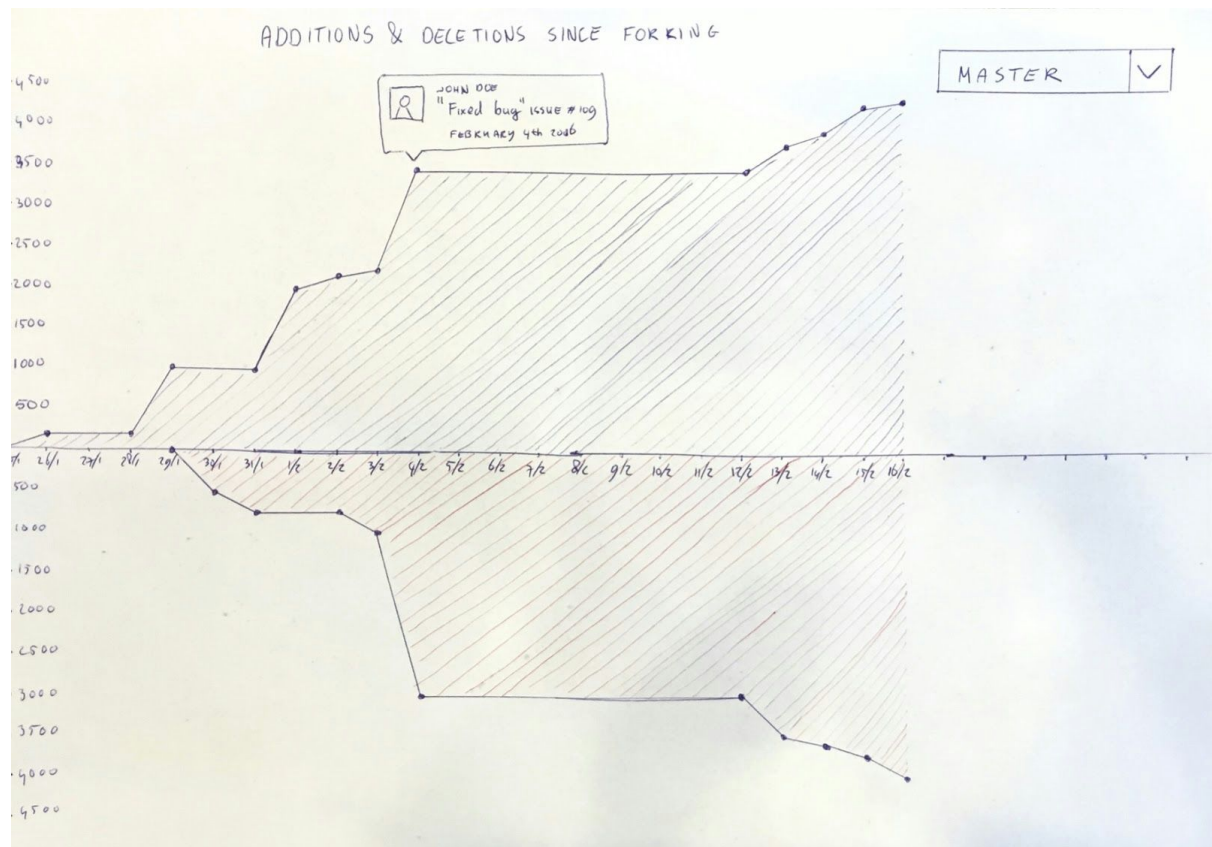
The visualization shows the additions or removals of lines of code relative to a chosen fork in a github project. Clicking on any point in the graph will show a brief description of the changes that were made at that point. The numbers on the y-axis shows the amount of additions and removals in steps that are automatically adjusted to the size of the project. The x-axis shows the date in steps that are automatically adjusted to the amount of time.

2.
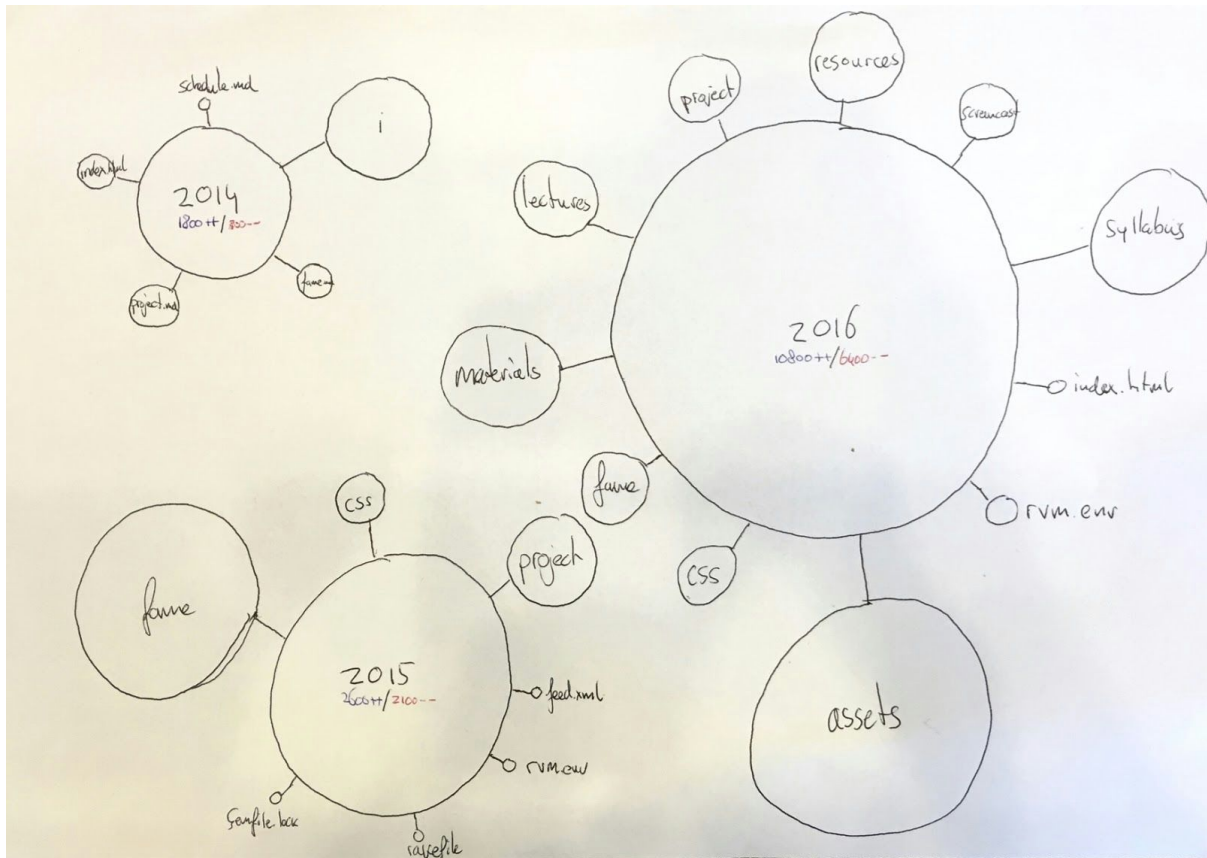
This visualization visualizes the different files and folders of the project and the amount of changes made to them. A larger ball indicates more changes to the files in the folder (absolute). The subfolders of each folder are linked to their main folder in a corona. The visualization should be interactive: The balls are clickable and will act like a file explorer when clicked upon, displaying the underlying folders and files. Maybe it would also be nice to indicate the people who changed things in a file by making the file clickable as well.

The size of the circles should adapt automatically to the available area on the screen. The numbers in the central folders can indicate the "scale" (in additions and deletions).

PART 3 - GROUP REFLECTION

We have chosen the first visualisation. Because the visualisation shows very clear the amount of deletions and additions in the project code.

The visualization is very useful to see how much the project has evolved since you branched off of it. The drop down menu at the top could be used to see the total amount of changes per branch or the total amount of changes for the entire project.

The first visualization does not however indicate where the changes have been made. In order to figure that out, it would be nice if, for example, the axes were clickable to interactively change the graph: one could set a certain value of the x-axis as a starting point to see the amount of changes made from that point on, and then have the ability to enter a visualization like the second one, in which the changes to the different subfolders are indicated.