

Data processing

Design 2: github commit graph

Groep B3:
Jaap Nieuwenhuizen
Daniël Hansen
Arnoud Smit
Marije Dekker

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.

Relevante attributen

Voorbeeld 1: users, branches, forks, commit, colour, timeline

Voorbeeld 2: users, owners, branches, forks, timeline.

Mogelijk relevante attributen

Voorbeeld 1: hoeveelheid van het werk wat is gecommitt.

Voorbeeld 2: hoeveelheid van het werk wat is gecommitt.

Lijst met alle attributen in netwerk

Voorbeeld 1: users, branches, forks, commit, colour, timeline

Voorbeeld 2: users, branches, forks, commit, colour, timeline

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

User roles: contributors, owners

Deze rollen hebben andere doelen.

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

contributor

De contributor is vooral bezig met zijn eigen code/opdracht, vaak in zijn eigen branch. Daarnaast is het voor de contributor fijn om te weten wat er met zijn branch/commits wordt gedaan.

owner

De owner kan met behulp van de visualizatie makkelijker overzicht houden over de repository. Daarnaast kunnen de verschillende branches in de gaten gehouden worden. De owner heeft daarom ook behoefte aan de kwantiteit van attributen over de hele repository heen.

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

De owner, geprioriteerd ↓

attributen:

1. branches
2. forks
3. commit
4. colour
5. timeline
6. amount

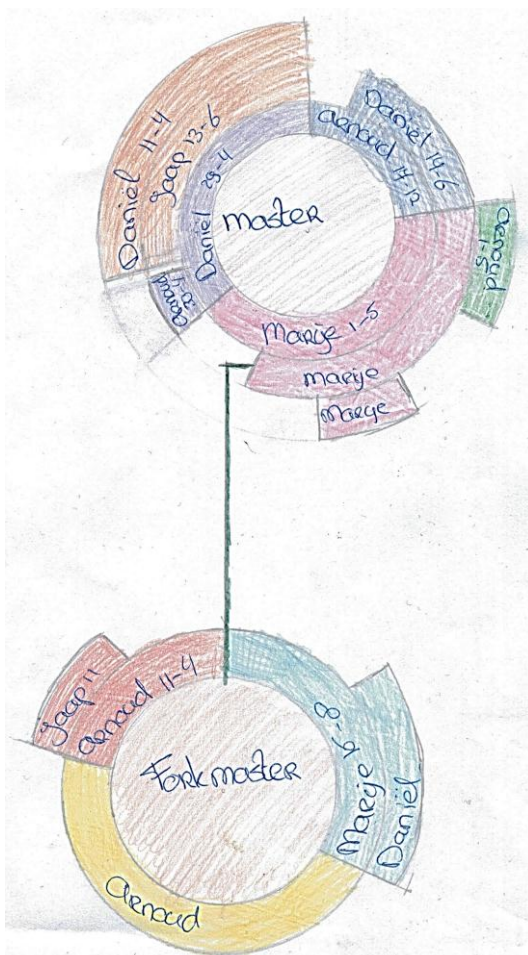
taken:

1. overzicht, attributen kunnen aflezen
2. branches in de gaten houden, de samenhang er tussen kunnen zien
3. de masterbranch kunnen herkennen

PART 2 - Sketching

Design two alternative visual representations for the Github Network. You should design for an interactive system, i.e., you should not assume that you have to fit all content onto paper. You can refer to Heer's article Reading 3 for the examples of network visualisations, browse D3 website, and feel free to use any other sources.

Schets 1

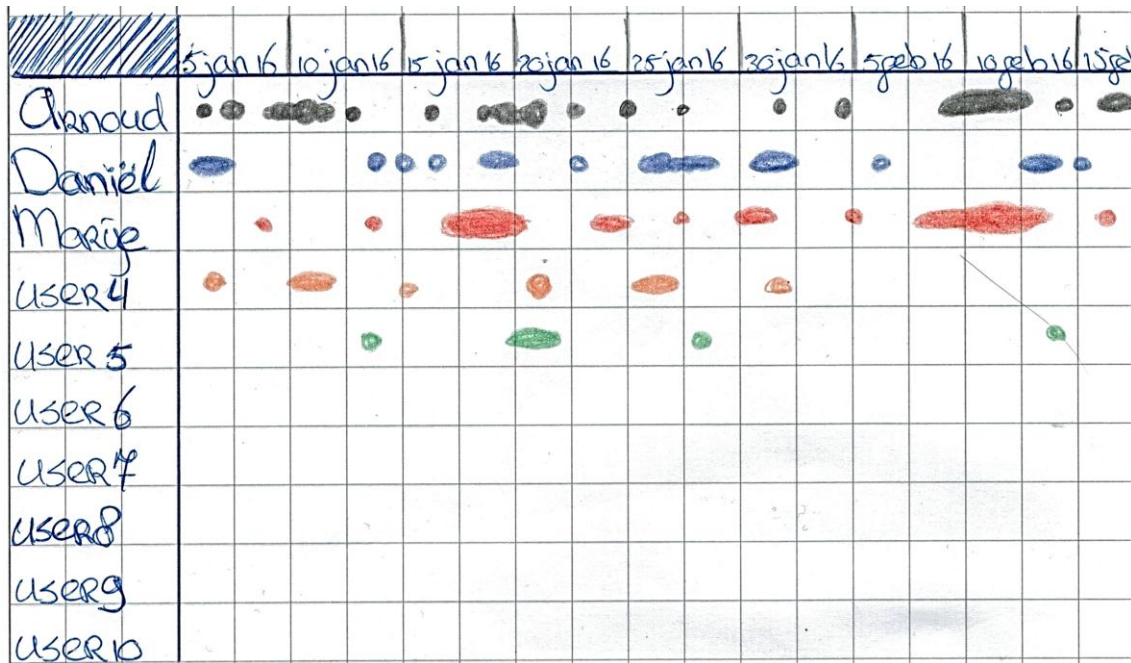


Wanneer er een commit wordt aangeklikt, verschijnt in het midden een percentage van de hoeveelheid en de gebruikers gegevens. Elke commit stapelt zich op binnen een branch (vertegenwoordigd door dezelfde kleur). De breedte van een ring geeft aan hoe groot de commit is. Zodra een andere kleur op de ring wordt gestapeld dan is er sprake van een merge.

De merge kan dus ook over meerder kleuren plaatsvinden omdat er dan meerdere branches met elkaar zijn gemerged.

Schets 2

Hierbij is elke rij een user. De bellen aan de rechterkant geven aan hoeveel er gecommited is en wanneer. Als je in zoomed dan worden de bellen losse punten. Omdat meerdere losse punten samen een grote vlek vormen. Elke gebruiker krijgt een eigen kleur. Op de hoofd-as staan de data weergegeven. Deze zijn nu per 5 dagen. Als er verder wordt ingezoomd dan gaan dit dus per dag. Als er wordt uitgezoomd dan worden deze intervallen groter.



PART 3 - Group Reflection

Take your analysis and ideas and discuss it again. Can you find a consensus? Come up with one visualization that you agree is ideal. Reflect on the process that you went through to come up with your final design (refer to this week's lecture: Process for inspiration on design process).

In de visualisatie van github is geen rekening gehouden met de hoeveelheid van de commit. In onze visualisaties hebben wij geprobeerd dit er wel in te verwerken. Uiteindelijk hebben wij besloten dat de eerste schets het beste en het meeste weergeeft. Aan de rechterkant staat deze schets opnieuw weergegeven. Daarnaast geeft deze grafiek een duidelijk samenhang tussen de verschillende repo's en de onderlinge relatie (forks). Toch is de progressie voor elke branch en de progressie tussen repo's makkelijk af te lezen. De grafiek is simpel in zijn eenvoud maar toch volledig en voorziet de gebruiker van veel informatie.

