

Week 4 – Github Commit Graph

Lotte Slim – Saskia Valstar – Jan Maarten de Vries – Sjors Witteveen – Laura Veerkamp

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.

Attributes: gebruikers, tijdsschaal, masterbranch, sidebranches (kleurcodering), mergetags, commits

De networks verschillen van elkaar in aantal synchroon lopende branches, het aantal gebruikers en tijdsduur.

De grootte van de verandering is niet weergegeven. De ene gebruiker commit veranderingen viermaal daags, terwijl de andere dit eens in de aantal weken doet. Er is geen verschil te zien in de schaal of de aard van de commit (toevoeging, verwijdering, verandering, om hoeveel files het gaat).

Hoezeer de branches van elkaar verschillen is niet af te leiden uit de figuur. Branches splitsen zich af in de y-richting. Totdat er opnieuw gecommited wordt worden branches uit elkaar getrokken, onafhankelijk van de inhoud.

- Are there different roles, i.e., different types of users who might want to achieve different things? Write a list of user roles. Think about which tasks a user of your visualization might want to achieve. Write down a list of tasks.

Ontwikkelen

Controleren

Communiceren

Leren

Overnemen (met goedkeuring)

Proces in de gaten houden

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

We kiezen voor communiceren, omdat het schrijven van code in een groep gecoördineerd dient te verlopen om dubbel werk te voorkomen en ieders bijdrage overzichtelijk te houden. Het is dan bevorderlijk om inzichtelijk te maken wat er in elke commit gebeurt. Gedetailleerde commitbeschrijvingen kunnen ook helpen.

Part 2 - Sketching

- Decide on which visual variable to use for which attributes of the visualizations. Consider the strengths and weaknesses of visual variables that were mentioned in Carpendale's article Reading 2 (also briefly discussed in this week's lecture: Process). Use the strongest visual variable for the most important attributes of the data.

In dit geval is position de belangrijkste visual variable, want daarmee laat je zien vanuit welke branch er wordt geüpload naar de andere branch en welke commits er op welk moment worden gedaan en dat is de voornaamste functie van deze grafiek.

- Do you think it is necessary to represent every single commit as a separate node? Could you think of ways to aggregate this?

Je zou er ook voor kunnen kiezen om de commits alleen weer te geven in het interactieve deel van het netwerk, dat je alleen de commits naar de masterbranch weergeeft als node.

- Do you think that every contributor needs a "row", as on the default network view on github? Could you think of a smarter way to summarize those?

Het is wel overzichtelijk als je per row een contributor hebt, maar je zou ook met kleur de verschillende contributors kunnen aangeven.

- Is a node-link diagram the appropriate representation? Or should you consider alternative graph representations?

Wij denken dat voor de hoeveelheid informatie die in het netwerk verwerkt moet worden, een node-link grafiek daarvoor geschikt is. Er gebeurt veel in de grafiek, daarom vinden wij de tijdlijn een goede basis om het overzicht te bewaren.

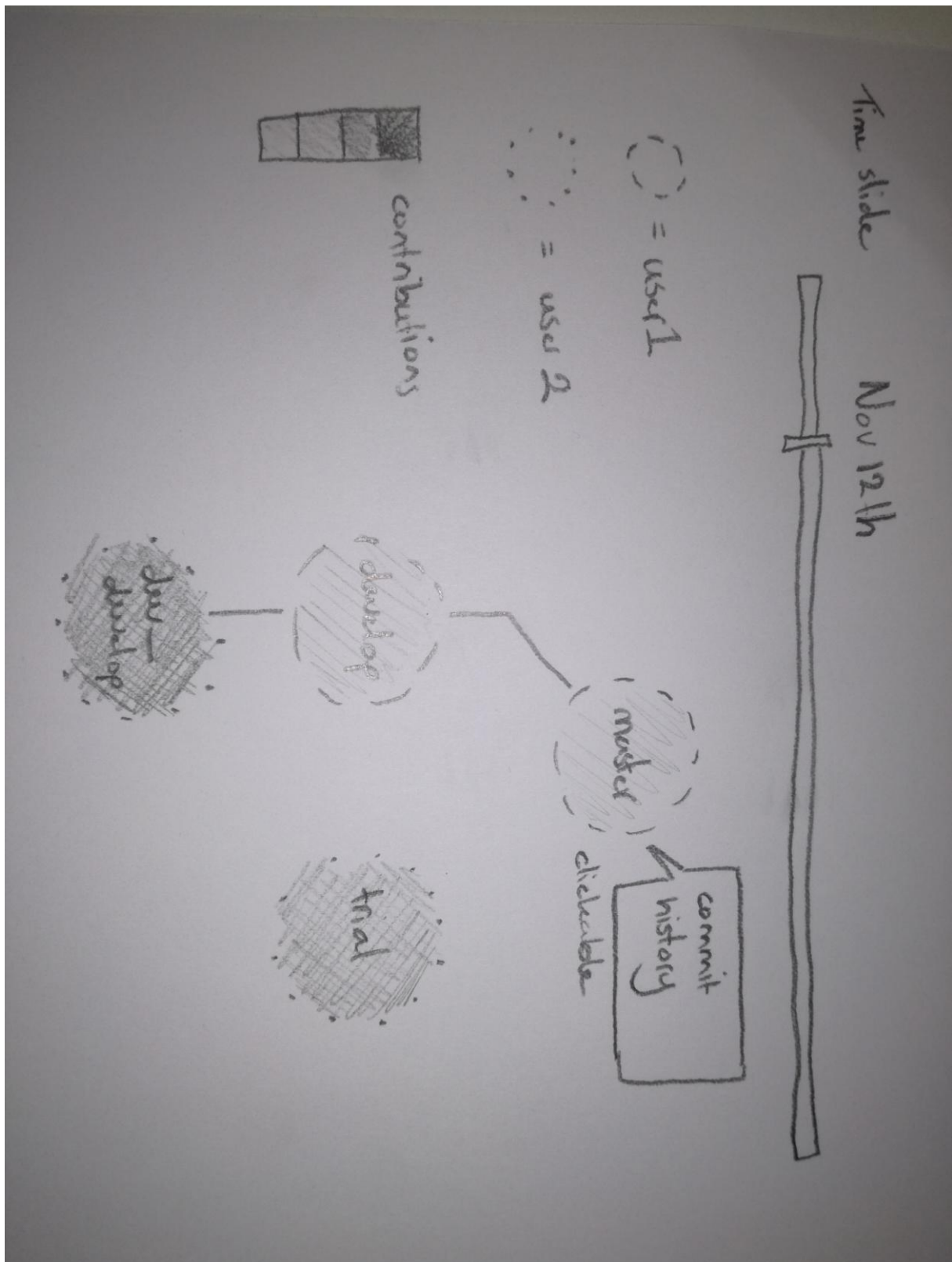
PART 3 - Group Reflection

- 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).

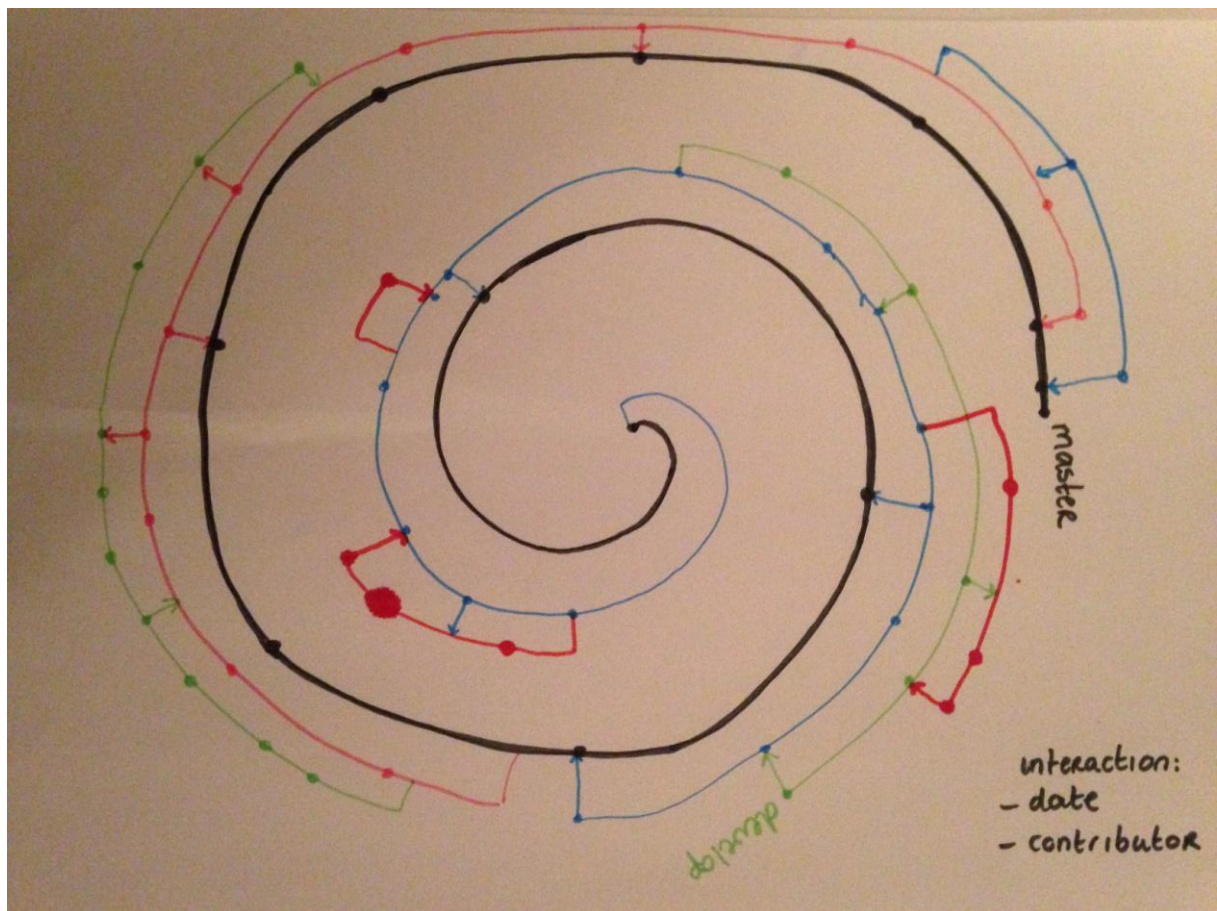
Uiteindelijk kiezen we voor de grafiek met het spiraalvormige netwerk (sketch 2), aangezien die het meest weg heeft van het oude netwerk. In alle mogelijke ideeën liepen we er tegenaan dat het combineren van de tijd met het weergeven van het verloop van de branches en de commits moeilijk weer te geven is. Daarom is het een aantal rijen (branches) in een netwerk met de tijdlijn een goede basis om mee te beginnen. We waren het snel eens dat de contributor niet heel belangrijk is om weer te geven, dus dat kan in het interactieve deel worden gezet.

We vonden vooral dat de grafiek zoals hij op GitHub wordt gebruikt al een goede manier is om de data te laten zien, daarom vonden wij het lastig om met een heel nieuw idee te komen. Het enige nadeel van de grafiek op github was dat het lastig was om overzicht te krijgen omdat je omdat er maar een klein tijdsbestek wordt weergegeven op je beeldscherm. In een spiraal vorm kun je op 1 blaadje al een heel groot deel van de grafiek weergeven.

Sketch 1



Sketch 2



Sketch 3

