

Design week 4

Groep 4

Sanne Meijering	10783709
Tim Molleman	10587306
Jelle Mul	11402148
Shan Li Nio	6222420
Misha Paauw	10054154

PART 1 – ANALYSIS

De grafiek in voorbeeld 1 heeft veel branches. De belangrijkste branch wordt hier gebruikt om het gedeelte dat nog in ontwikkeling is (de development branch) van de master branch te scheiden. De meeste andere branches worden gemerged met de development branch of met elkaar. In voorbeeld 2 wordt alles direct gecommited naar de master, waardoor de branches klein zijn.

Eigenschappen die in de visualisatie staan:

- Commits
- Commit size
- Commit content (on click)
- Date of action
- Person who did something
- Branches en merges
- Fork acties

Eigenschappen die missen in de visualisatie

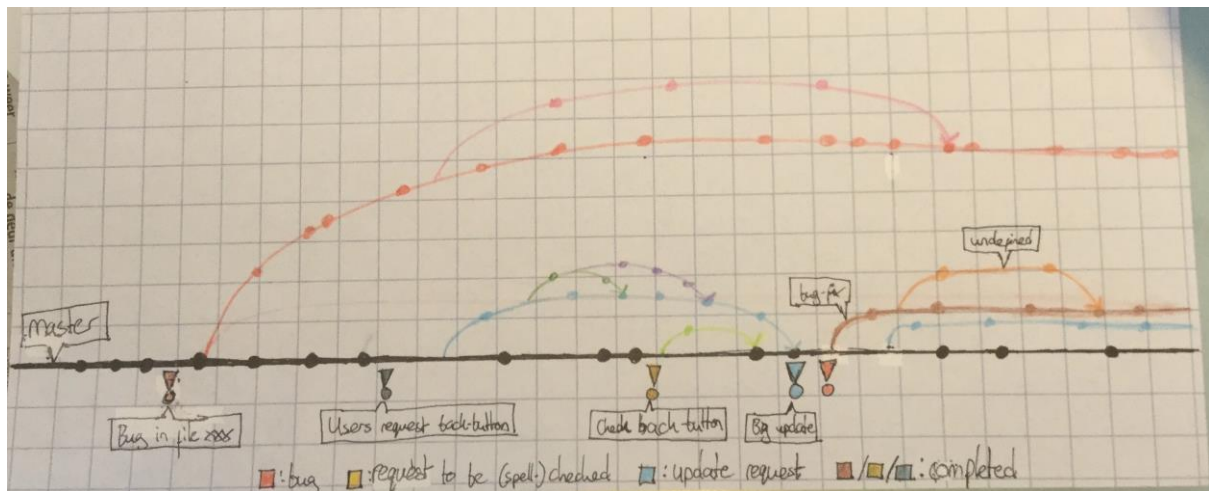
- Conflicten

Er kunnen verschillende typen gebruikers zijn. Bijvoorbeeld ontwikkelaars die nieuwe functies maken, en anderen die wellicht meer bezig zijn met het fixen van bugs en typfouten. Ook zou er 1 'leider' kunnen zijn, die uiteindelijk beslist wat er daadwerkelijk gepubliceerd wordt.

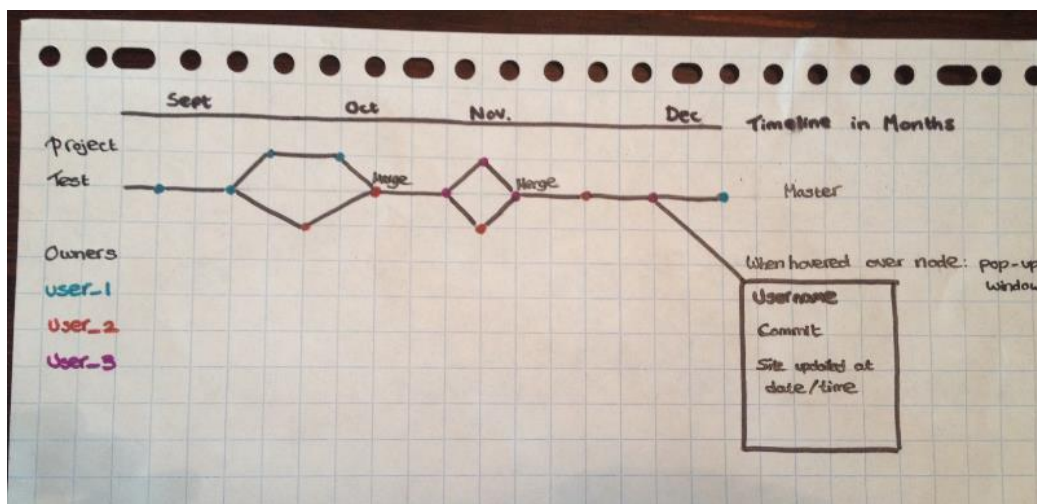
Lijst van dingen die een user wilt doen:

- Historie van een project bekijken
- Kijken of er nog branches zijn die nog niet gemerged zijn.

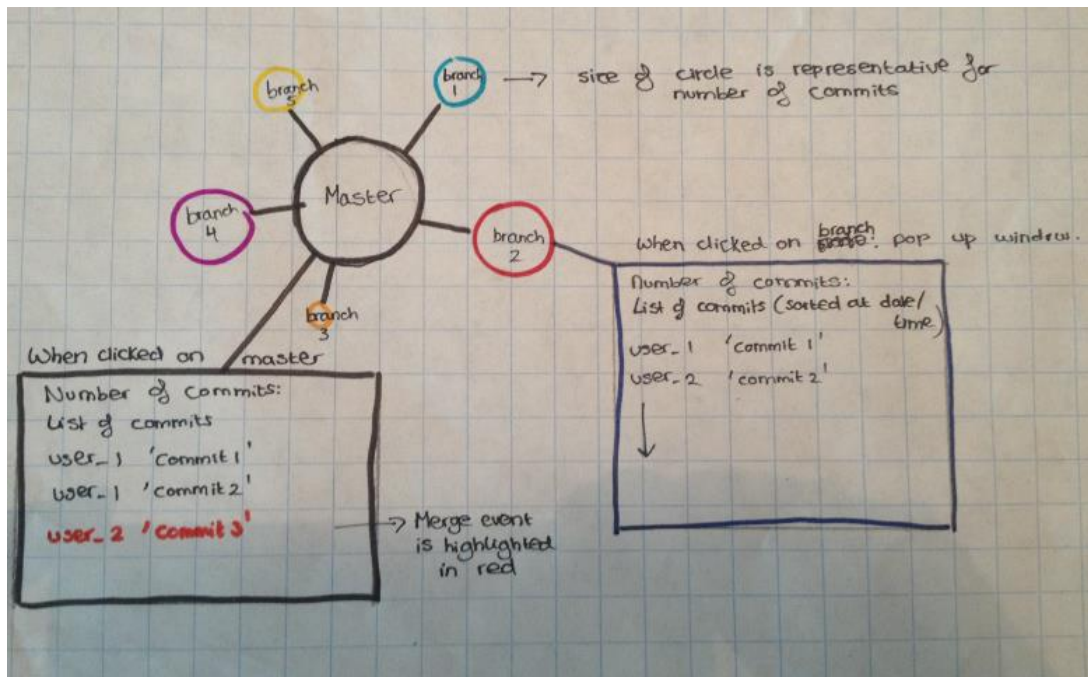
PART 2 – SKETCHING



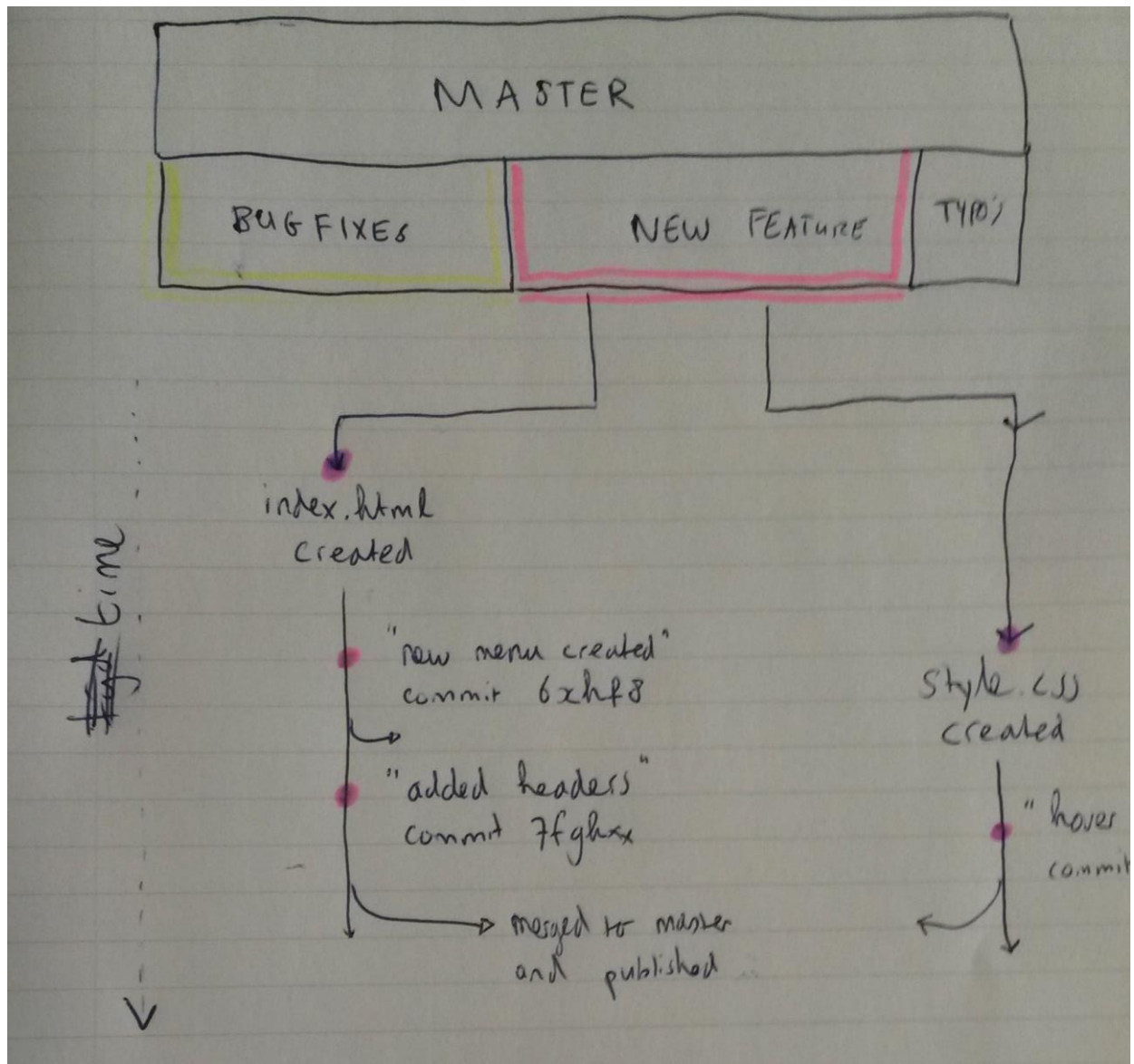
Visualisatie 1. Branches worden aangegeven met lijnen en commits worden aangegeven met punten op de lijnen. De master branch is de onderste lijn en de langst bestaande branch is de bovenste lijn. Lijnen kruisen elkaar alleen als oude branch gemerged wordt terwijl een nieuwe branch nog bestaat. Onder de masterbranch staan uitroepstekens die aangeven waar bugs gevonden zijn, waar code en tekst moet worden nagekeken en waar updates van het programma gevraagd worden. Elk type uitroepsteken heeft een eigen kleur en wordt donkerder als aangeven wordt dat het probleem verholpen is.



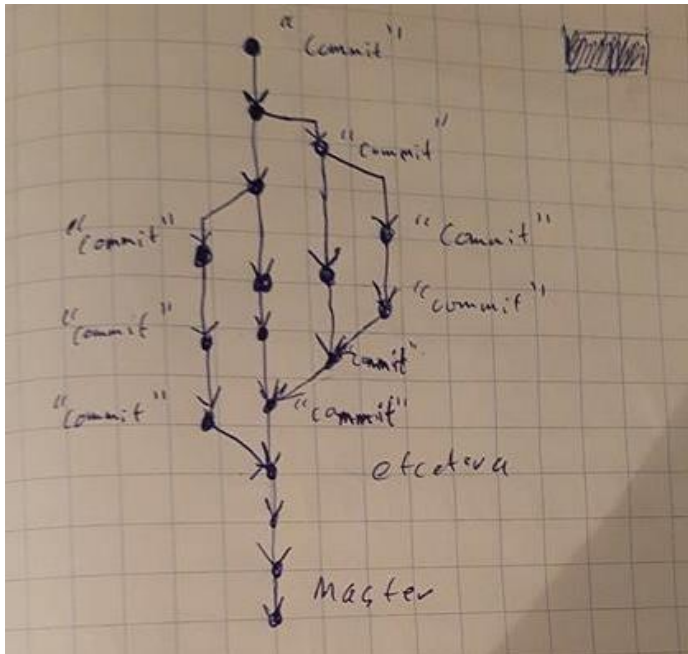
Visualisatie 2. De master branch wordt weergegeven in zwart. Door verschillende kleuren te gebruiken voor de verschillende users, wordt visueel duidelijk welke user welke commit heeft gedaan. Verder wordt er duidelijk aangegeven wanneer users tegelijkertijd aan bestanden werken (het splitsen van de master branch) en wanneer er sprake is geweest van merging (het bijeenkomen van de 2 afsplitsingen). Dit is belangrijk aangezien de meeste bugs ontstaan bij merge events. Bovenaan is een globale tijdlijn te zien in maanden en voor een gedetailleerd overzicht van de datum/tijd en van de commit kan er over de nodes gehovered worden, zodat er een pop-up window verschijnt met aanvullende informatie. Deze visualisatie is geschikt voor users die het overzicht willen houden over het verloop van het project en over de verschillende werkzaamheden van alle users.



Visualisatie 3. De master branch wordt weergegeven in het zwart, terwijl elke andere branch wordt weergegeven in een andere kleur. De grootte van de cirkels is representatief voor het aantal commits dat er in deze branch gedaan zijn. Op deze manier worden niet alle commits individueel weergegeven, maar worden ze gecombineerd in 1 visualisatie. Wanneer er op een branch geklikt wordt, verschijnt er een pop-up window, waarin het aantal commits nog een keer wordt aangeduid (deze keer in tekst) en een lijst van alle commits inclusief de users. De "merge" commits worden gehighligt in rood, waardoor ze makkelijk te vinden zijn. Daarom is deze visualisatie vooral geschikt voor het oplossen van bugs, aangezien erbij merge events de grootste kans op het ontstaan van bugs is. Omdat het verloop van de verschillende branches niet heel belangrijk is in dit geval, worden deze dan ook schematisch afgebeeld, maar wel op een manier zodat het duidelijk is of er veel of weinig commits in deze branch zijn gedaan.



Visualisatie 4. Bovenstaande representatie laat de master branch zien, welke wordt onderverdeeld in een aantal branches. De breedte van de branches laat de grootte daarvan zien (in aantal commits of in het aantal regels code). Wanneer er op een branch geklikt wordt opent de bekende boomstructuur uit de voorbeelden. Echter zijn het geen branches, maar de verschillende files. Het is namelijk belangrijk om te kunnen zien in welke file er is gewerkt. Ook staan er kleine pijltjes, die aangeven wanneer een file/commit naar de master is gemerged en dus gepubliceerd.



Visualisatie 5. Deze visualisatie is redelijk vergelijkbaar met de oorspronkelijke visualisatie, echter wordt hier gebruik gemaakt van een verticale boomstructuur. Hierbij fungeert de master als stam en de verschillende branches als takken. Door het draaien van de visualisatie is het naar onze mening in 1 oogopslag duidelijk wat de structuur is van de repository.

PART 3 - Group Reflection

Aan het begin zijn we als groepje samengekomen om te discussiëren over de opdracht te proberen alternatieve visualisaties te bedenken. Uiteindelijk hebben we ervoor gekozen om apart allemaal één visualisatie te bedenken. Dit heeft ervoor gezorgd dat er uiteindelijk meer uiteenlopende visualisaties gecreëerd zijn dan wanneer we als groep over een alternatieve visualisatie hadden nagedacht. In het laatste geval zou er waarschijnlijk veel aandacht besteed zijn aan één soort alternatieve visualisatie, wat nu niet het geval is.

Uiteindelijk is er door ieder groepslid een stem uitgebracht op de alternatieve visualisatie die hij/zij het beste vond. Visualisatie 1 kreeg uiteindelijk de meeste stemmen. Deze visualisatie lijkt veel op de originele github, aangezien wij het uiterlijk van het origineel met verschillende gekleurde branches en nodes voor elke commit erg overzichtelijk vonden. Deze visualisatie heeft echter als toevoeging dat gevonden bugs, updates, en stukken code die meer aandacht nodig hebben kunnen worden aangegeven met uitroepstekens onder de masterbranch. Op deze manier is het in één oogopslag duidelijk waar de problemen zijn en of deze problemen al verholpen zijn.