

Design week 4.

PART 1 – ANALYSIS

In voorbeeld 1 wordt gebruik gemaakt van branches. Deze branches zijn om bepaalde ontwikkelingsstadia van elkaar te scheiden. Er wordt gemerged met de master als men tevreden is de ontwikkelings. In voorbeeld 2 wordt alles direct gecommited naar de master.

Eigenschappen die erin staan:

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

Eigenschappen die erbij zouden kunnen:

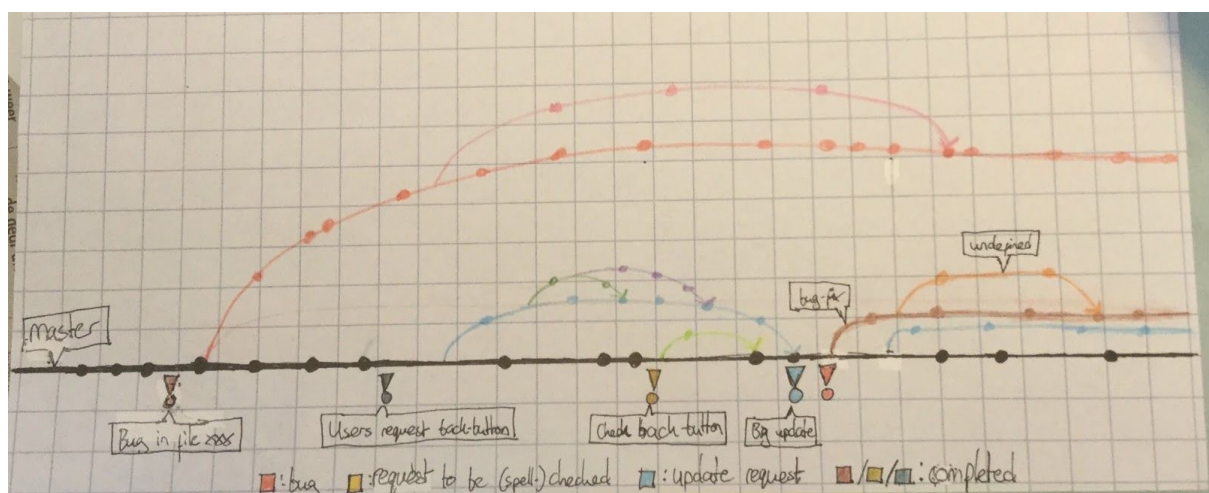
- Conflicten

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

Lijst van dingen die een user wilt doen:

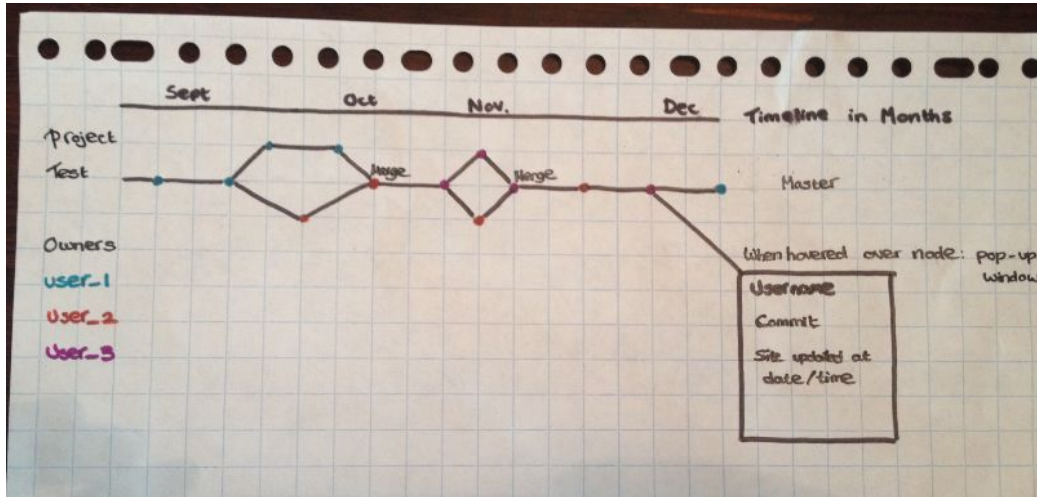
- Historie van een project bekijken
- Kijken of er nog branches zijn die nog niet gemerged zijn.

PART 2 – SKETCHING

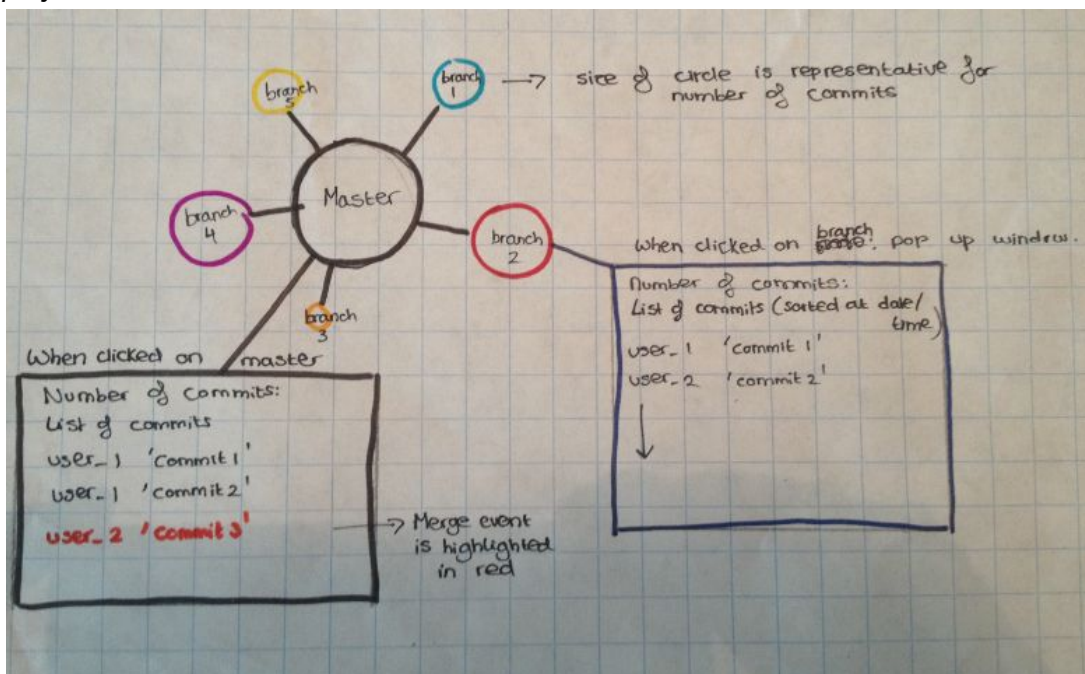


Visualisatie 1. Branches worden aangegeven met lijnen en commits worden aangegeven met punten op de lijnen. De masterbranch is de onderste lijn en de langst bestaande branch is de bovenste lijn. Lijnen kruisen elkaar als oude branch gemerged wordt terwijl een nieuwe

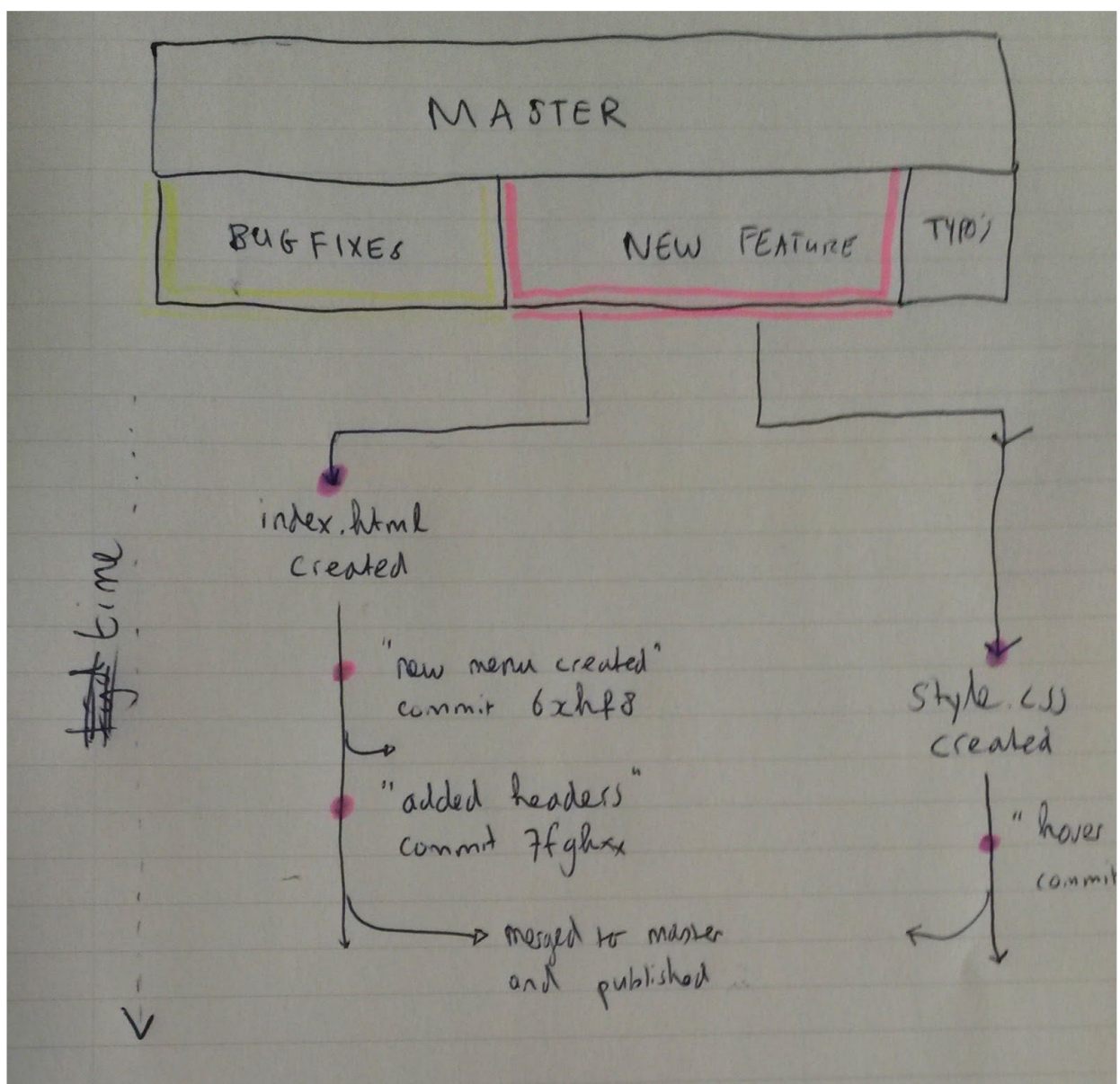
branch nog bestaat. Onder de masterbranch staan uitroepstekens die aangeven waar bugs gevonden zijn, waar code en tekst moet worden nagekeken en waar updates van het programma gevraagd worden. Elk type uitroepsteken heeft een eigen kleur en wordt donkerder als aangeven wordt dat het probleem verholpen is.



Visualisatie 2. De masterbranch 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 masterbranch) 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.

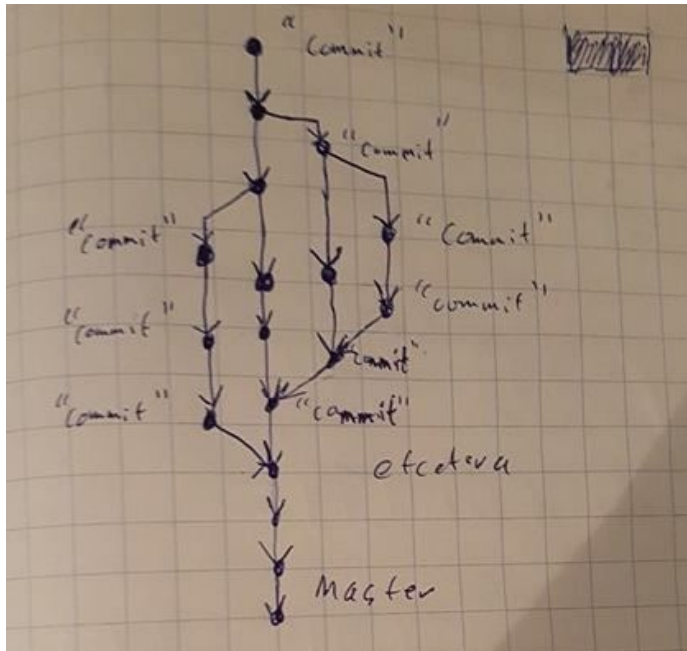


Visualisatie 3. De masterbranch wordt weergegeven in het zwart, terwijl elke andere branch wordt weergegeven in een andere kleur. De grootte van de cirkels is representatief voor het aantal commits dat er in deze branch gedaan zijn. Op deze manier worden niet alle commits individueel weergegeven, maar 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. Wanneer op de masterbranch wordt geklikt verschijnt wederom een pop-up window met een lijst van commits, maar in deze lijst worden de "merge" commits gehighlight in rood. Daarom is deze visualisatie vooral geschikt voor het oplossen van bugs, aangezien er bij de merge events de grootste kans op bugs is. Aangezien 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 zien, welke wordt onderverdeeld in een aantal branches. De breedte van de branches laat de grootte daarvan zien (in aantal

commits of mogelijk het aantal regels code?). Wanneer er op een branch geklikt wordt opent de bekende boomstructuur. Echter zijn het geen branches, maar de verschillende files. Ik vind namelijk dat het belangrijk is 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.

PART 3 - Group Reflection

Allereerst zijn we als groepje samengekomen om te discussiëren over de opdracht, daarnaast hebben we geprobeerd 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 uitroepetekens onder de masterbranch. Op deze manier is het in één oogopslag duidelijk waar de problemen zijn en of deze problemen al verholpen zijn.