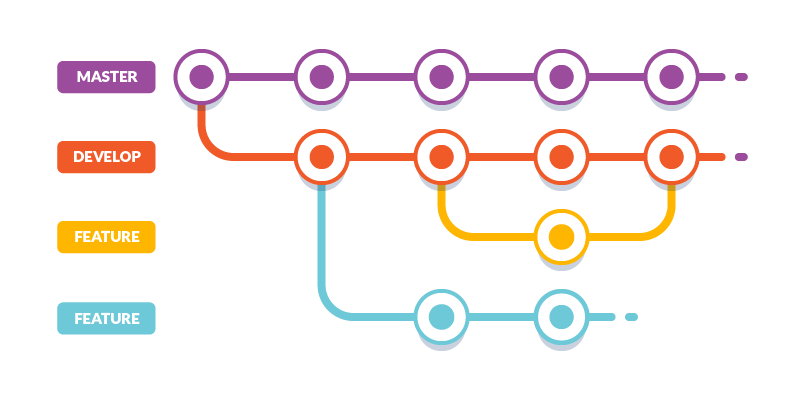
23-9-2022

Muijsenberg, Reno R.F.

Fontys

Git Flow

Research



Inhoud

[Aanleiding 2](#_Toc114820392)

[Wat is Git Flow 2](#_Toc114820393)

[Branches 2](#_Toc114820394)

[Main 2](#_Toc114820395)

[Develop 2](#_Toc114820396)

[Feature 2](#_Toc114820397)

[Release 2](#_Toc114820398)

[Hotfix 2](#_Toc114820399)

# Aanleiding

# Git Flow

Git Flow is de oudste strategie voor het beheren van je code met git. Het zorgt ervoor dat je je code goed kan onderhouden op bepaalde niveaus. Het wordt veel gebruikt door open source projecten zodat ze goed code die gepusht wordt kunnen monitoren.

## Branches

In deze strategie heb je twee hoofd branches: de main branch en de develop branch. Verder heb je nog een aantal kleinere branches die hier ook mee gebruikt worden zoals: feature, release en hotfix.

### Main

De main branch is bedoeld als stabiel release dit is dan ook de branch waar geen fouten op mogen staan. Vaak wordt vanaf deze branch direct deployed naar de live server.

### Develop

Je werkt op de develop branch wanneer dit werk op een stabiel punt komt wordt het gemerged in main branch met een release nummer.

### Feature

De feature branch wordt gebruikt voor wordt gemaakt vanaf de develop branch voor een kleine feature en wordt wanneer af weer terug gemerged in develop branch.

### Release

De release branch is bedoeld voor wanneer je de code op het punt komt dat het gereleased kan worden deze branch wordt dan gebruikt voor het klaarmaken/ laatste bug fixen voordat het in de main branch wordt gemerged.

### Hotfix

De hotfix branch wordt gebruikt voor het oplossen van urgente problemen. Deze branch wordt gemaakt vanaf de main branch en hier ook rechtstreek in terug gemerged.

## Nadelen

Een van de nadelen van deze werkwijze is dat wanneer een feature te lang op zijn/haar eigen feature branch bevind je merge conflicten kan krijgen omdat het niet meer synchroon loopt met de reset van het project. Verder kan deze methode voor projecten die continuouse intergration hebben een slomere oplossing zijn.

https://rewind.com/blog/git-branching-strategies-explained/