

Version Control

TADRALA, PIOTR P.P. 487080

Inhoudsopgave

Versiebeheer	2
Voorwoord	2
Agile werkwijze	2
Trello	2
Github	3
Github Projects	3
Product Backlog	4
Sprint Backlog	4
In Progress.....	4
In Review	4
Done.....	4
Backlog	5

Versiebeheer

VERSIE	DATUM	OPMERKING
1.0	14-01-23	Eerste concept van het document.

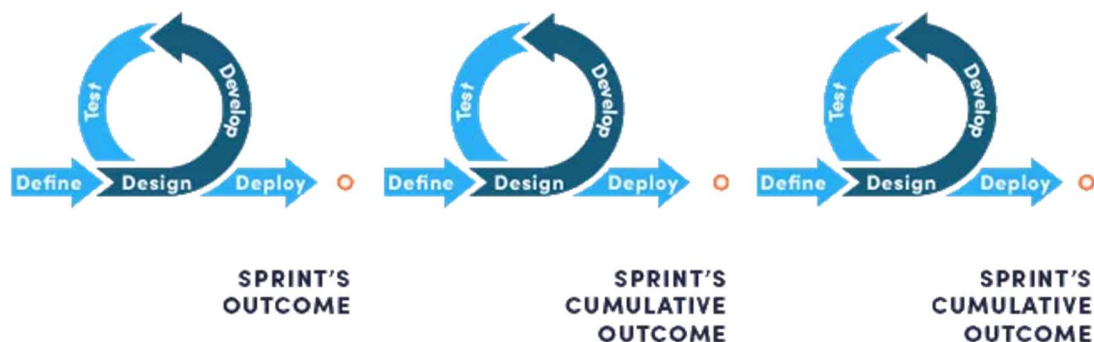
Voorwoord

Aan het begin van ons project hebben enkele van ons een workshop gevolgd over het werken met de Agile-methode. Hierdoor besloten we om deze werkwijze toe te passen in ons project. In eerste instantie hadden we twijfels of we direct aan de slag zouden gaan met de Agile-methode aangezien we al halverwege de sprint waren, maar uiteindelijk besloten we om te wachten tot de huidige sprint was afgerond (*vandaar dat we het ook sprint 0 hebben genoemd*) en pas vanaf de volgende sprint met Agile te beginnen.

Agile werkwijze

Agile is een manier om projecten te beheren en te plannen. Het is bedoeld om snel en flexibel te kunnen reageren op veranderingen die zich tijdens het project kunnen voordoen. Dit in tegenstelling tot de traditionele manier van projectmanagement waarbij alles van tevoren wordt gepland en vastgelegd.

Agile werkt met korte iteraties van een paar weken, genaamd sprints. Tijdens elke sprint werkt het team aan een aantal taken uit de backlog, die de lijst is van dingen die nog gedaan moeten worden. Aan het einde van elke sprint wordt er een demonstratie gegeven van wat er is bereikt en wordt er gekeken naar wat er de volgende sprint gedaan moet worden. Zo blijft het project voortdurend in beweging en kan het snel aangepast worden als er veranderingen zijn.

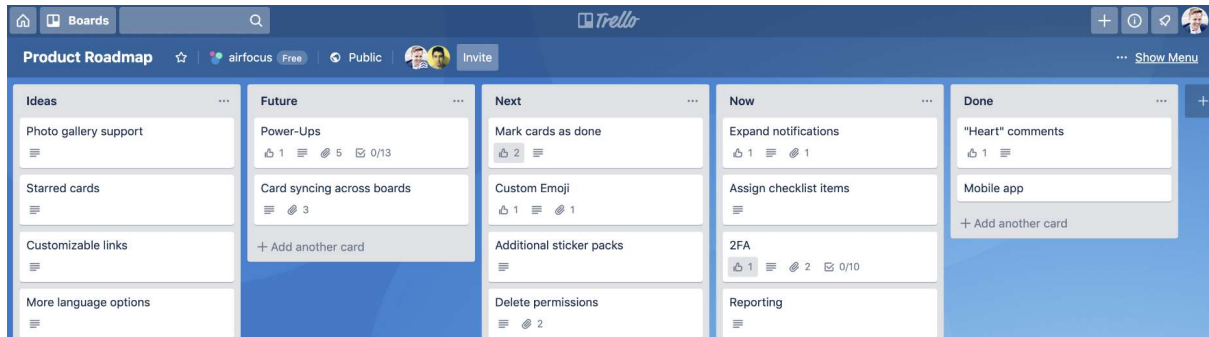


Trello

In eerste instantie hebben we besloten om onze backlog in Trello bij te houden. Trello is een hulpmiddel waarmee een gebruiker een bord kan aanmaken waarop het gehele proces van een project wordt vastgelegd.

Per bord kan je kolommen definiëren die aangeven in welke fase een onderdeel zich bevindt. Bijvoorbeeld 'To Do', 'In Progress' en 'Done'. Vervolgens kunnen items worden toegevoegd. De omschrijving van elk item geeft aan waar het om gaat en eventuele sub-onderdelen, zoals een checklist, kunnen worden toegevoegd. Zodra je begint met een onderdeel, wordt het item verplaatst naar 'In Progress' en zodra het klaar is naar 'Done'.

Na een paar weken besloten we echter om de Trello te verlaten, niet omdat Trello niet aan onze eisen voldeed voor backlog-tracking, maar omdat het ook mogelijk was om het proces te volgen in GitHub Projects. Aangezien we GitHub al gebruikten voor versiebeheer van onze code, vonden we het handiger om het backlog ook in GitHub bij te houden.



Github

GitHub is het grootste wereldwijde platform voor versiebeheer van code. Aangezien ons team volledig uit software developers bestond, was ieder van ons meer of minder bekend met GitHub. Het was dus een logische keuze om GitHub te gebruiken voor het versiebeheer van ons project.

Om te beginnen met GitHub moesten we eerst een repository aanmaken. Een repository functioneert als een soort container waarin de code wordt opgeslagen. Een repository kan worden opgedeeld in branches. In eerste instantie begin je met een branch genaamd 'Main'. Als iemand een aanpassing maakt, kan diegene een pull request aanvragen voor die aanpassing. Voor die aanpassing wordt dan een nieuwe branch aangemaakt. De beheerder van de repository moest vervolgens beslissen of de request wordt gemerged met de Main branch of afgewezen. Zodra een request wordt gemerged, kunnen de andere teamleden de nieuwe code 'pullen' vanuit de Main branch. Op deze manier is het mogelijk om samen aan een codebase te werken.



Github Projects

GitHub Projects is een tool die binnen GitHub beschikbaar is voor het beheren van projecten. Het is een visuele manier om taken en takenlijsten te beheren, gekoppeld aan de code in een repository. Het maakt het mogelijk om de voortgang van een project te volgen en te organiseren.

Met GitHub Projects kan je borden aanmaken en organiseren in kolommen (*Hetzelfde principe als eerder benoemde Trello toepassing.*). Deze kolommen kunnen bijvoorbeeld 'To Do', 'In Progress' en 'Done' zijn. De gebruiker kan vervolgens kaarten toevoegen aan deze kolommen, waarop

bijvoorbeeld een taak of een bug is beschreven. De kaarten kunnen ook labels, assignees en deadlines hebben. Dit maakt het gemakkelijk om te zien wie wat doet en wanneer iets af moet zijn.

Wij hebben onze boord in de volgende categorieën verdeeld:

Product Backlog

Onderdelen die aan het eind van het project besproken en afgerond moeten zijn.

Sprint Backlog

Onderdelen die aan het eind van de sprint besproken en afgerond moeten zijn

In Progress

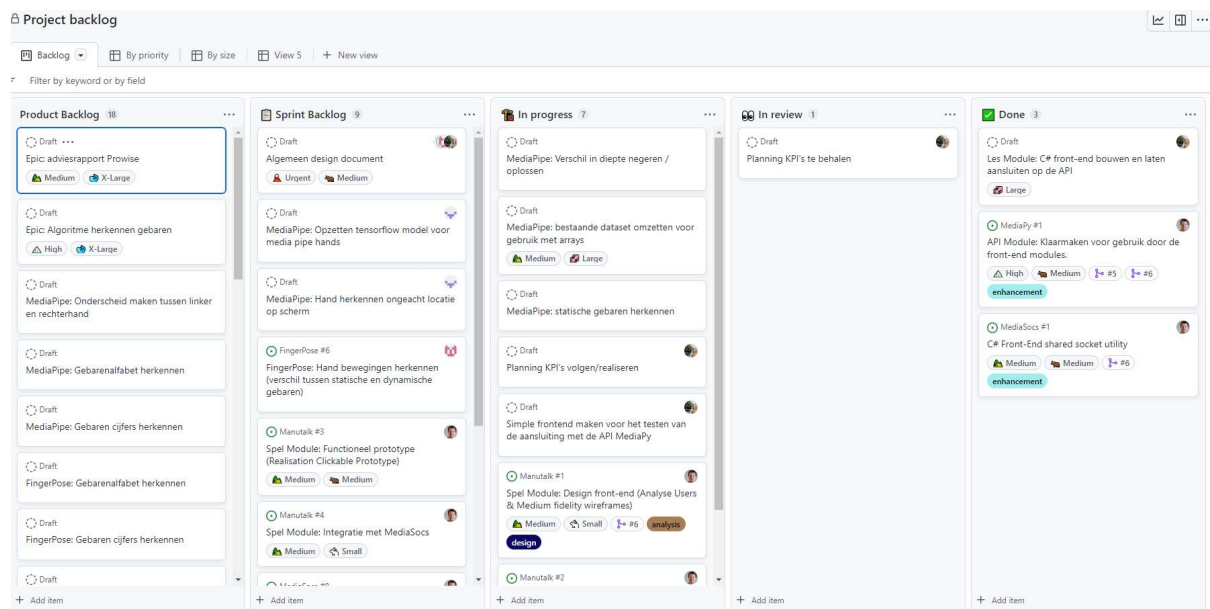
Onderdelen waar iemand aan werkt

In Review

Onderdelen die uit de huidige sprint die afgerond zijn maar nog niet besproken.

Done

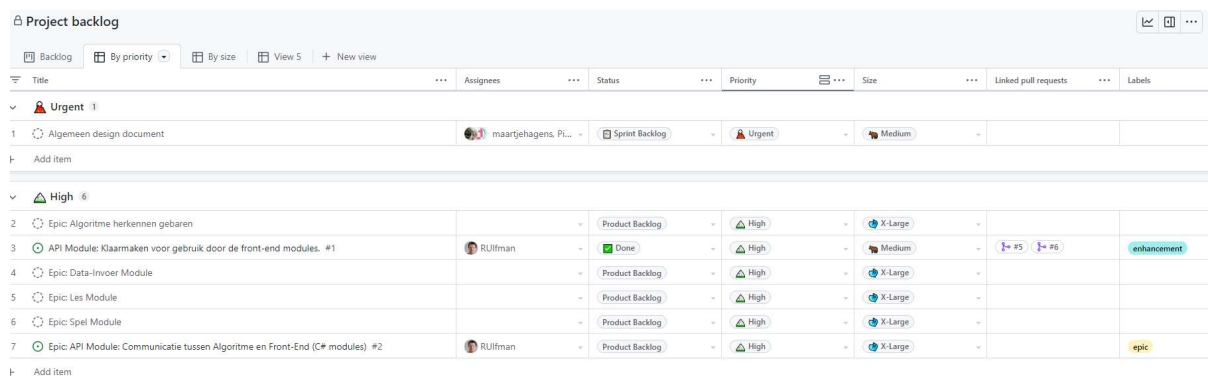
Volledig afgeronde onderdelen.



Ook hebben we 2 custom properties toegevoegd. Namelijk 'priority' en 'Size':

- **Priority**
 - Urgent
 - High
 - Medium
 - Low
- **Size**
 - X-Large
 - Large
 - Medium
 - Small
 - Tiny

Dankzij deze properties was het voor ons makkelijk om een goed overzicht te houden voornamelijk bij het inplannen van een nieuwe backlog. Ook geeft Github een mogelijkheid om op basis van custom properties te sorteren waardoor het voor ons mogelijk was om snel een overzicht te krijgen van de meest dringende onderdelen.



The screenshot shows a GitHub Project backlog titled "Project backlog". It features a filter bar with options like "By priority" and "By size". The backlog is organized into sections based on priority: "Urgent" and "High". Each item in the backlog is a row with columns for Title, Assignees, Status, Priority, Size, Linked pull requests, and Labels. The "Urgent" section contains one item: "Algemeen design document" assigned to "maartjehagens, PI..." with status "Sprint Backlog", priority "Urgent", and size "Medium". The "High" section contains seven items, mostly assigned to "RUIJfman", with various statuses (e.g., "Done", "Product Backlog") and sizes (e.g., "X-Large", "Medium"). Some items have labels like "enhancement" or "epic".

Title	Assignees	Status	Priority	Size	Linked pull requests	Labels
Urgent						
1 Algemeen design document	maartjehagens, PI...	Sprint Backlog	Urgent	Medium		
High						
2 Epic: Algoritme herkennen gebaren		Product Backlog	High	X-Large		
3 API Module: Klaarmaken voor gebruik door de front-end modules. #1	RUIJfman	Done	High	Medium	#5 #6	enhancement
4 Epic: Data-Invoer Module		Product Backlog	High	X-Large		
5 Epic: Les Module		Product Backlog	High	X-Large		
6 Epic: Spel Module		Product Backlog	High	X-Large		
7 Epic: API Module: Communicatie tussen Algoritme en Front-End (C# modules) #2	RUIJfman	Product Backlog	High	X-Large		epic

Backlog

Aan het begin van elke sprint, hebben we afgesproken om elkaar te ontmoeten en minstens een kwartiertje te besteden aan het bekijken en bijwerken van onze backlog. Hierbij bespraken we de voortgang van iedereen en verplaatsten we voltooide taken naar de juiste kolom in ons GitHub Projects bord.

Na deze besprekingen, gingen we verder met het bepalen van de volgende stap voor ons project. Hierbij gaven we prioriteit aan de taken met de hoogste prioriteit en creëerden we een nieuwe planning voor de komende sprint (*Sprint Backlog*). Dit proces heeft ons geholpen om efficiënt te werken en ons project op schema te houden.