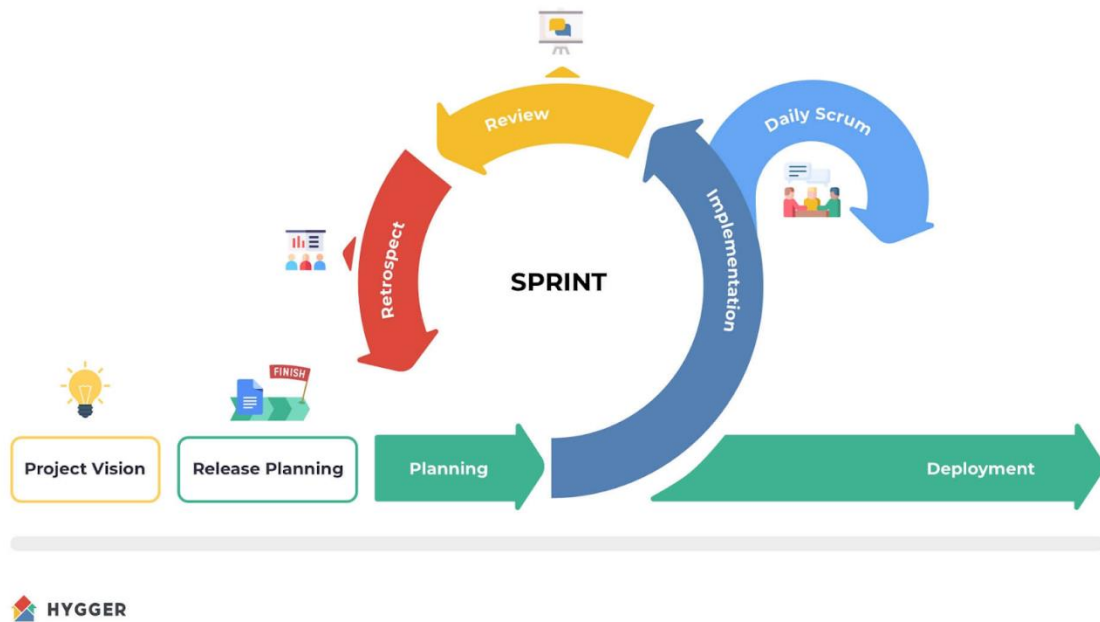


Scrum documentatie - crossplatform development



Door: Sadek al Mousawi en Renas Khalil

Datum: 20-01-2025

Vak: Crossplatform Development

Inhoudsopgave

Inleiding	3
Scrum	4
Azure DevOps.....	8
Sprint review.....	13
Retrospectives.....	15
Nawoord	17

Inleiding

In dit document beschrijven we hoe wij de Scrum-methodiek hebben toegepast tijdens het project Cross-Platform Development (2024/2025). Het doel van dit document is om inzicht te geven in onze werkwijze, de gebruikte tools en hoe deze hebben bijgedragen aan een succesvolle projectuitvoering.

Scrum, als een flexibele en iteratieve aanpak, was enorm belangrijk voor onze samenwerking. Het stelde ons in staat om op een gestructureerde en efficiënte manier samen te werken.

Naast Scrum hebben we gebruikgemaakt van verschillende ondersteunende tools en methodieken om ons werk te organiseren en te communiceren:

Azure DevOps: Voor het beheren van sprints, taken, en voortgangsrapportages.

Discord: Als primair communicatiemiddel.

In de volgende secties van dit document zullen we gedetailleerd ingaan op hoe deze methodieken en tools ons hebben geholpen bij het plannen, uitvoeren en succesvol afronden van het project. Hierbij delen we onze ervaringen, uitdagingen en de lessen die we onderweg hebben geleerd.

We laten een paar voorbeelden van ons Azure-board zien, maar we raden aan om het hele board zelf te bekijken. Hierin vind je de volledige uitwerkingen van verschillende onderdelen, zoals alle user stories, een uitgebreide wiki, alle retrospectives en userstories.

<https://ictwf.visualstudio.com/Cross-Platform-04>

Ook kunt u in de map zelf de Azure-export vinden met de volledige backlog. Deze wordt niet in dit document behandeld.

Scrum

Way of Working:

We hanteren de Fibonacci-reeks voor het inschatten van tijd. Daarbij staat 1 story point gelijk aan 1 uur. De Fibonacci-reeks die we volgen, is als volgt: 0, 1, 2, 3, 5, 8, 13, enzovoort.

Scrum:

Voordat een sprint begint, stellen we altijd een sprintdoel vast. In het sprintdoel beschrijven we welke epics en user stories we willen afronden in de aankomende sprint.

Tijdens de sprint maken we gebruik van dagelijkse stand-ups. Tijdens deze meetings bespreken we wat we gaan oppakken uit de backlog voor de huidige sprint en welke uitdagingen we tegenkomen.

Aan het einde van de sprint leveren we altijd een Minimal Viable Product (MVP) op aan de stakeholders. Dit geeft hen de mogelijkheid om de huidige voortgang van het project te beoordelen en feedback te geven.

Na de oplevering volgt een sprint retrospectief. Hierin evalueren we wat goed ging, wat beter kon, en hoe we deze verbeteringen willen toepassen in de volgende sprint.

Deze cyclus herhalen we voor elke sprint en vormt de kern van onze werkwijze.

Sprintdoelen:

Sprint 1 +2:

1. Goed begin aan de documentatie: Het opzetten en starten van de projectdocumentatie om een duidelijke richtlijn te hebben gedurende het hele project.
2. GitHub-opzet: De repository structureren en organiseren op GitHub, inclusief branches, commit-standaarden en workflows.
3. Design + feedback op designs: Het creëren van een eerste ontwerp en het verzamelen van feedback om de ontwerpen te verbeteren en af te stemmen op de wensen van de stakeholders.
4. Login-functionaliteit: Het implementeren van een gebruiksvriendelijke en veilige loginmodule.
5. Teams ophalen (fetchen): Functionaliteit toevoegen om teams dynamisch vanuit de backend te laden en weer te geven.

Sprint 3 + 4:

1. Teams aanmaken: Functionaliteit toevoegen waarmee gebruikers nieuwe teams kunnen creëren.
2. Evenementen aanmaken: Mogelijkheid bieden aan gebruikers om evenementen te plannen en op te slaan.
3. Loginpagina: Ontwerpen en implementeren van een intuïtieve en veilige inlogpagina.
4. Registratiepagina: Een gebruiksvriendelijke registratiepagina ontwikkelen voor nieuwe gebruikers.
5. Teams CRUD: Volledige CRUD-functionaliteit (Create, Read, Update, Delete) voor het beheren van teams.
6. Evenementen CRUD: Volledige CRUD-functionaliteit voor het beheren van evenementen.

Sprint 5:

1. Documentatie afgerond: Het documenteren van alle projectdetails, processen en functionaliteiten volledig voltooid.
2. User Experience (UX) / User Interface (UI) afgerond: Het ontwerpen en optimaliseren van een gebruiksvriendelijke interface en fijne gebruikerservaring volledig afgerond.
3. Agenda-/roosterpagina afgerond: Ontwikkeling en implementatie van een pagina voor het beheren van agendas of roosters volledig voltooid.

Definition of Done:

Voor de developer:

Een taak of functionaliteit wordt als "klaar" beschouwd door de developer wanneer:

Code is geschreven en voldoet aan de vereisten van de user story: De functionaliteit moet precies werken zoals beschreven in de user story en voldoen aan de acceptatiecriteria.

Unit tests zijn geschreven en slagen: Voor de nieuwe code zijn unit tests geschreven die alle belangrijke paden en functionaliteiten afdekken. Deze tests moeten zonder fouten slagen.

Code is gereviewed en goedgekeurd: De code is door een andere developer gereviewed en voldoet aan de standaarden voor leesbaarheid, onderhoudbaarheid en efficiëntie.

Geen openstaande bugs of issues: De functionaliteit moet zonder fouten werken volgens de specificaties, en er mogen geen openstaande kritieke bugs of problemen zijn.

Code is geïntegreerd in de main branch: De code moet correct geïntegreerd zijn in de hoofdcodabase, zonder merge-conflicten.

Documentatie is bijgewerkt: Indien nodig, is de documentatie van de code of de technische documentatie bijgewerkt met de wijzigingen.

Voor de gebruiker:

Een taak of functionaliteit wordt als "klaar" beschouwd door de gebruiker wanneer:

De functionaliteit werkt zoals verwacht zonder problemen: De gebruiker moet de functionaliteit kunnen gebruiken zoals beschreven in de user story, zonder onverwachte fouten of beperkingen.

De gebruikersinterface is intuïtief en gebruiksvriendelijk: Voor functies met een UI is de interface begrijpelijk en eenvoudig te gebruiken, zonder dat er een complexe uitleg of training nodig is.

De functionaliteit is grondig getest in een realistische omgeving: De functionaliteit is getest in een staging- of productieomgeving en werkt zonder problemen, zowel functioneel als qua prestatie.

Er is geen verdere feedback of verbeterpunten: De gebruiker heeft de functionaliteit getest en goedgekeurd, en er is geen aanvullende feedback of verzoek om wijzigingen meer nodig.

Algemeen voor het team:

Een taak of functionaliteit is pas volledig "klaar" wanneer:

Alle acceptatiecriteria van de user story zijn voldaan.

Functionaliteit is getest en gedocumenteerd: Alle testen (unit, integratie, acceptatie) zijn doorlopen en alle relevante documentatie is bijgewerkt.

De feature is gedemonstreerd aan de stakeholders: Het team heeft de functionaliteit gedemonstreerd aan de relevante stakeholders of product owner, en deze is goedgekeurd zonder verdere aanpassingen.

De functionaliteit is vrijgegeven voor gebruik: De functionaliteit is beschikbaar gesteld in de productieomgeving en klaar voor gebruik door eindgebruikers.

Azure DevOps

Wiki:

In onze wiki behandelen we het volgende onderwerpen we zullen deze in het kort uitleggen maar adviseren om deze zelf door te nemen op onze azure omgeving.

<https://ictwf.visualstudio.com/Cross-Platform-04>

Prioteiten:

1. Must have
2. Should have
3. Could have
4. Won't have

We hebben onze prioriteiten gestructureerd van hoog naar laag, zodat het voor ons duidelijker is. Prioriteit 1 heeft de hoogste prioriteit, gevolgd door 2, 3, en 4.

Story points:

Story points worden gebruikt om de complexiteit en de hoeveelheid werk van een user story te schatten. Het helpt het team om inzicht te krijgen in de inspanning die nodig is om een taak af te ronden. Hierbij wordt rekening gehouden met factoren zoals:

1. Complexiteit: Hoe moeilijk is de taak technisch of inhoudelijk?
2. Onzekerheid: Hoeveel onbekende factoren of risico's zijn er?
3. Omvang: Hoe groot is de hoeveelheid werk?

Het team gebruikt een relatieve schaal, zoals de **Fibonacci-reeks** (1, 2, 3, 5, 8, etc.), om taken te scoren en gezamenlijk conclusie te schatten.

Definition of done:

De Definition of Done is een duidelijke checklist die bepaalt wanneer een user story echt als "klaar" wordt beschouwd. Dit helpt consistentie en kwaliteit te waarborgen. Een voorbeeld van

Kano model:

Het Kano-model wordt gebruikt om klantbehoeften te analyseren en te categoriseren, wat helpt bij het prioriteren van productfeatures. Door het Kano-model toe te passen, kunnen we bepalen welke features prioriteit moeten krijgen tijdens de ontwikkeling.

Productgoal & Productvisie:

Productgoal: Het concrete doel dat het team wil bereiken met het product.

Productvisie: Een bredere, inspirerende beschrijving van wat het product op lange termijn wil bereiken.

Een goed gedefinieerde productgoal en -visie geven richting aan het team en zorgen ervoor dat alle activiteiten bijdragen aan het grotere plaatje.

Stakeholder analyses & matrix:

Een stakeholderanalyse helpt bij het identificeren en begrijpen van de verschillende belanghebbenden in een project.

Hierbij worden stakeholders geclassificeerd op basis van:

- Invloed: Hoeveel invloed heeft de stakeholder op het project?
- Belang: Hoe belangrijk is het project voor de stakeholder?

Deze informatie wordt vaak weergegeven in een stakeholdermatrix met vier vakken:

1. Hoge invloed, hoog belang: Actief beheren (bijvoorbeeld belangrijke investeerders).
2. Hoge invloed, laag belang: Tevreden houden (bijvoorbeeld externe partners).
3. Lage invloed, hoog belang: Informeren (bijvoorbeeld eindgebruikers).
4. Lage invloed, laag belang: Monitoren (bijvoorbeeld indirect betrokkenen).

Door de stakeholderanalyse en -matrix te gebruiken, kunnen we gericht communiceren en prioriteiten stellen.

Voorbeeld UserStories:

De userstories zijn met de hulp van https://www.wikiwand.com/en/User_story#Usage gemaakt, hiervoor heb de template Connextra template gekozen.

<div> <div></div> <div>31227 Teams pagina Front-end</div> </div>	
<div> <div>RK</div> <div>Renas Khalil</div> </div>	8
Area Path	Cross-Platform-04
Iteration Path	Iteration 4
Created By	<div> <div>RK</div> <div>Renas Khalil</div> </div>
Priority	2
<div>Front-end</div>	

USER STORY 31227

31227 Teams pagina Front-end

Renas Khalil

0 comments

Front-end

X

+

Statg

Active

Area

Cross-Platform-04

Reason

Implementation started

Iteration

Cross-Platform-04\Iteration 4

Description

Als gebruiker wil ik een teams pagina zien, zodat ik een overzicht krijg van alle teamleden en hun rollen binnen de organisatie, en eenvoudige contact kan opnemen met specifieke personen of teams.

Acceptance Criteria

1. De teams pagina bevat een overzicht van alle teams en teamleden, gestructureerd.

2. Elk teamlid wordt weergegeven met naam, profielfoto.

3. Er is een zoek- en filterfunctie beschikbaar om snel specifieke teamleden te vinden.

4. De lay-out past zich aan verschillende schermformaten aan (responsive design).

Discussion

Planning

Story Points

8

Priority

2

Risk

Classification

Value area

Business

Personas affected

Closed

31222

Homepage Back-end

RK

Renas Khalil

10

Area Path

Cross-Platform-04

Iteration Path

Iteration 4

Created By

RK Renas Khalil

Priority

1

Back-end

Voorbeeld Backlog:

Cross-Platform-04 Team

Backlog Analytics + New Work Item View as Board Column Options ...

	Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path	Tags
+	1	User Story	Teams pagina Front-end	Active	8	Business	Cross-Platform-04\Iteration 4	Front-end
	2	User Story	Rooster pagina Back-end	New	20	Business	Cross-Platform-04\Iteration 1	Back-end
	3	User Story	Rooster pagina Front-end	New	10	Business	Cross-Platform-04\Iteration 1	Front-end
	4	User Story	Registratie Pagina Front-end	New	18	Business	Cross-Platform-04\Iteration 1	Front-end
	5	User Story	Homepagina - Front-end	New	12	Business	Cross-Platform-04\Iteration 1	Front-end
	6	User Story	LogIn Pagina - Front-end	New	8	Business	Cross-Platform-04\Iteration 2	Front-end
	7	User Story	Documenteren van Mockups	Active	8	Business	Cross-Platform-04\Iteration 2	Documentatie
	8	User Story	Design Mockups	Active	12	Business	Cross-Platform-04\Iteration 2	Documentatie
	9	User Story	Creëren van UML-diagrammen Description	Active	8	Business	Cross-Platform-04\Iteration 1	Documentatie

Voorbeeld Retrospectives:

Retrospectives Cross-Platform-04 Team

Board History Retrospective iteration 2 Prime Directive Team Assessment Collect Group Vote Act

What went well

+ Add new feedback

Documenteren / opzetten van azure
13 December 2024 at 02:59
#1

Voorbereiding van presenteren
13 December 2024 at 02:59
#2

Communicatie
13 December 2024 at 02:59
#3

Samenwerking
13 December 2024 at 02:59
#4

What didn't go well

+ Add new feedback

Beter inschatting qua hoe lang / groot een user storie is en duurt te doen
13 December 2024 at 02:59
#1

daily stands up
13 December 2024 at 02:58
#2

eerder / optijd beginnen
13 December 2024 at 02:59
#3

Mocht u alles in het geheel willen zien dan adviseren wij om de **Azure DevOps** export door te nemen voor de backlog, userstories, epics.

En om <https://ictwf.visualstudio.com/Cross-Platform-04> alles in het geheel te zien.

Sprint review

Sprint Review

Tijdens onze sprint reviews hebben we onze voortgang regelmatig gepresenteerd en feedback verzameld van stakeholders en teamleden. Hieronder beschrijven we per sprint de belangrijkste punten van onze reviews:

Sprint 1 en 2:

De sprint reviews voor deze sprints vonden plaats op school, waar we onze eerste voortgang presenteerden. We lieten het volgende zien:

- **Documentatie:** De opzet en voortgang van de projectdocumentatie.
- **Design:** De eerste ontwerpen en visuele concepten van het project.
- **Azure Scrum Board:** Een overzicht van onze sprints, user stories, en toegewezen taken in Azure DevOps.
- **Minimum Viable Product (MVP):** Het eerste functionele product met basisfunctionaliteiten, waaronder:
 1. Een veilige en gebruiksvriendelijke inlogfunctionaliteit.
 2. Het dynamisch laden (fetchen) van teams vanuit de api.
 3. De mogelijkheid voor gebruikers om nieuwe teams aan te maken.

Sprint 3 en 4:

Ook deze reviews vonden op school plaats en volgden hetzelfde format. Tijdens deze reviews presenteerden we:

- **Updates van eerdere sprints:** Verbeteringen en uitbreidingen van de functionaliteiten die tijdens sprint 1 en 2 zijn ontwikkeld.
- **Nieuwe functionaliteiten:**
 1. Het aanmaken van evenementen door gebruikers.
 2. Het weergeven van evenementen gekoppeld aan specifieke gebruikers.
- **Focus op codekwaliteit:** Tijdens deze sprints hebben we extra aandacht besteed aan:
 1. Het opsplitsen van complexe code in kleinere, overzichtelijke modules.
 2. Het schrijven van duidelijke en gestructureerde comments.
 3. Het bijwerken van technische documentatie om de leesbaarheid en onderhoudbaarheid van onze code te verbeteren.

Sprint 5:

De review van sprint 5 draaide vooral om reflectie en afronding onderling. We hebben:

1. Een overzicht gegeven van alles wat we hebben ontworpen, ontwikkeld en opgeleverd gedurende het project.

2. Een uitgebreide terugblik gedaan op ons proces en besproken wat goed ging en wat beter kon.
3. Besproken hoe we eventuele verbeterpunten in toekomstige projecten kunnen oppakken.

De sprint reviews waren essentieel voor het ontvangen van waardevolle feedback, het valideren van onze voortgang en het verder verfijnen van onze aanpak.

Retrospectives

De retrospectives vonden meestal direct na de sprint reviews plaats. Tijdens deze sessies reflecteerden we op onze manier van werken en bespraken we wat goed ging, wat beter kon, en hoe we ons proces konden verbeteren voor de komende sprints. Deze retrospectives waren een belangrijk onderdeel van ons Scrum-proces omdat ze ons hielpen om transparant te blijven over de huidige stand van zaken en continu te verbeteren.

Doel en aanpak:

Tijdens elke retrospective hebben we structureel gekeken naar:

- **Wat goed ging:** Welke processen en samenwerking goed verliepen en bijdroegen aan het succes van de sprint.
- **Wat minder goed ging:** Obstakels, uitdagingen of inefficiënties die we tegenkwamen.
- **Hoe we konden verbeteren:** Concrete acties om onze werkwijze aan te passen en de prestaties te optimaliseren in de volgende sprint.

Belangrijke inzichten:

Een perfect voorbeeld van hoe de retrospectives bijdroegen aan onze verbetering is wat we ontdekten na sprint 1 en 2.

- **Uitdaging:** Tijdens deze retrospectives realiseerden we ons dat het project een enorm hoeveelheid documentatie en oplevering vereiste, meer dan we hadden ingeschat.
- **Aanpassing:** Op basis hiervan hebben we besloten om ons in de daaropvolgende sprints meer te focussen op het bijwerken en uitbreiden van onze documentatie en applicatie. Dit zorgde ervoor dat onze voortgang beter werd vastgelegd en overzichtelijker was voor alle stakeholders.

De retrospectives waren niet alleen nuttig om onze werkprocessen te verbeteren, maar ook om een open en eerlijke gesprek te voeren over onze samenwerking en voortgang. Hierdoor konden we als team beter presteren en hebben we geleerd om flexibel in te spelen op de uitdagingen die we onderweg tegenkwamen.

Hier een voorbeeld van de retroboard van sprint 2:

Retrospectives

Cross-Platform-04 Team

Board

History

Retrospective iteration 2

Prime Directive

Team Assessment

Collect

😊 What went well

+ Add new feedback

Documenteren / opzetten van azure

13 December 2024 at 02:59

#1

Vorbereiding van presenteren

13 December 2024 at 02:59

#2

Communicatie

13 December 2024 at 02:59

#3

Samenwerking

13 December 2024 at 02:59

#4

😞 What didn't go well

+ Add new feedback

Beter inschatting qua hoe lang / groot een user storie is en duurt te doen

13 December 2024 at 02:59

#1

daily stands up

13 December 2024 at 02:58

#2

eerder / optijd beginnen

13 December 2024 at 02:59

#3

16

Nawoord

We hopen dat we u met dit document een helder beeld hebben gegeven van hoe we de Scrum-methodiek hebben toegepast en geïntegreerd in ons project. Mocht u nog vragen hebben of feedback willen delen, dan staan we daar uiteraard voor open en horen we dit graag. Uw input is voor ons waardevol om onze processen en resultaten verder te verbeteren.