

H05: Ontwikkelstrategieën - Scrum & KanBan

1. Inleiding

Tijdens het opleidingsonderdeel Software Analysis hebben we het software-ontwikkelingsproces uitvoerig besproken. We hebben twee verschillende ontwikkelstrategieën, zijnde waterval en agile, bestudeerd op basis van de welke we het product konden ontwikkelen. Er zijn heel wat ontwikkelstrategieën die de principes van Agile toepassen. In dit hoofdstuk zullen we twee specifieke manieren in detail bespreken, zijnde Scrum en Kanban.

2. Scrum

2.1. Inleiding

Scrum is een open framework met eenvoudige regels die gebaseerd zijn op het principe van complex adaptieve systemen (CAS). We wensen als team te evolueren en hetzelfde proces steeds beter uit te voeren. Aangezien we volgens de principes van Agile werken zijn ook de twee belangrijkste eigenschappen, iteratief en incrementeel, bij scrum van primordiaal belang. Het inschatten van een sprint en de hoeveelheid werk die we kunnen uitvoeren tijdens een sprint zal nooit vanaf de eerste keer correct zijn. Maar hoe meer iteraties we uitvoeren, hoe preciezer en beter deze inschatting zal worden. Daarnaast bouwen we ook elke sprint verder aan de functionaliteiten van het systeem zodat deze incrementeel zullen uitgebreid worden. Deze manier van werken wordt dus vooral door ontwikkelteams gebruikt.

2.2. Regels

Indien een team scrum werkt zal het zich moeten houden aan volgende regels:

- Gesloten iteraties: we spreken tijdens de planning van een iteratie af hoeveel werk we zullen opnemen en wijken daar niet meer vanaf. De scope van een iteratie wijzigt niet meer tijdens zijn uitvoering ervan.
- Productiewaarde software: Na elke iteratie leveren we productiewaardige software op. Dit betekent dus dat er elke iteratie ook de nodige aandacht aan testen zal besteed moeten worden.
- Zelforganiserende en zelfreflecterende teams: Er is geen nood aan externe managers. Het team zelf organiseert zich en bepaalt dus welk werk ze in een iteratie zullen uitvoeren en hoe deze gepland word. Daarnaast leren ze ook van zichzelf en hun eigen werken om zo te evolueren doorheen de sprints.
- Prioriteiten: Het werk tijdens een sprint wordt uitgevoerd volgens een bepaalde prioriteit. Elke taak tijdens een iteratie heeft een prioriteit. Ook alle taken die ooit binnen het project dienen uitgevoerd worden, en dus niet noodzakelijk tijdens de huidige sprint, hebben een bepaalde belangrijkheid.-

2.3. Werking

Bij scrum zal bij de opstart van een project een “product backlog” aangemaakt worden. Deze bevat alle functionaliteiten die het eindproduct zal moeten bevatten. Deze taken zijn allen voorzien van een prioriteit. Bij de start van een nieuwe iteratie, sprint genaamd bij scrum, zal er een “sprint backlog” opgesteld worden. Deze bevat de taken die het team zal uitvoeren tijdens de volgende sprint. Ook deze taken hier zijn gesorteerd op prioriteit. Om de sprintbacklog op te vullen kan het

team een taak nemen uit de product backlog maar ook een onafgewerkte taak uit vorige sprints terug opnemen.

Na het opstellen van sprint backlog zal het team de sprint, typisch een periode van 2 tot 4 weken, uitvoeren waarbij ze dagelijks een “daily scrum” houden. Dit is een dagelijks overlegmoment waarin het team bespreekt wat de huidige status is.

Wanneer de iteratie afgelopen is volgt er een “sprint review” waarbij een demo aan de klant en de business gegeven wordt. Als afsluiter van de huidige iteratie wordt een “retrospectieve” gehouden om de werking van de huidige sprint te evalueren om te verbeteren naar volgende iteraties toe.

2.3.1. Sprint

Bij scrum spreken we van een sprint in plaats van een iteratie maar de betekenis is analoog. Typisch is de doorlooptijd van een sprint twee of drie weken. Het team doet er best aan om de lengte van een sprint constant te houden doorheen het project. Dit leidt tot een beter ritme omdat op die manier het inschatten meer zal verbeteren. Tijdens één sprint zal het product alle fasen van een iteratie van het software-ontwikkelingsproces doorlopen: analyse, ontwerp, programmeren, testen en integratie.

Gaan we uit van een sprint van 3 weken dan hebben we 21 dagen ter beschikking. Typisch zal deze sprint dan als volgt gepland worden:

- Dag 0: voorbereiding (laatste dag vorige sprint)
- Dag 1: productplanning en sprintplanning
- Dag 2-19: uitvoeren sprint + daily scrum
- Dag 20: Sprint review + retrospectieve

2.3.2. Voorbereiding sprint

Bij de start van het project wordt er dus een product backlog opgesteld. Deze bevat de features of taken aangereikt door de product owner of klant die in het uiteindelijke product moet zitten. Deze taken worden uniek geprioriteerd in de product backlog opgenomen. Bij de start van elke nieuwe sprint wordt tijdens de voorbereiding ervan steeds geëvalueerd of de product backlog nog steeds geldig is. Indien nodig worden taken toegevoegd of verwijderd of de prioriteit van een taak aangepast.

2.3.3. Planning sprint

Bij aanvang van de sprint zal op basis van de product backlog beslist worden welke items er zullen opgenomen worden. Deze ruwe inschatting van het aantal taken wordt uitgevoerd door het team zelf en dus niet door de product owner. Om dat mogelijk te maken moet het team wel over alle informatie beschikken om de taak te kunnen uitvoeren. Indien de producteigenaar geen antwoord kan formuleren op hun vragen om de ontbrekende informatie te verkrijgen dan zal het team die taak niet opnemen in die sprint. Als resultaat van deze oefening verkrijgt het team het sprintdoel.

Eens de taken gekozen zijn worden deze backlog items opgesplitst in uitvoerbare taken van 4 tot 16 uur en toegevoegd aan de sprint backlog. Het is belangrijk om de taken zo klein mogelijk te houden omdat dit zorgt voor een goed tempo binnen het team en er ook zorgt dat de inschatting zo weinig mogelijk afwijkingen vertoont ten opzichte van de uiteindelijke gepresteerde uren om die taak te verwezenlijken. Hoe groter de initiële workload van één taak is hoe meer kans er is op onderschatting van het team. De sprint backlog wordt opnieuw bepaald door enkel het team

aangezien we zelf-organiserend werken. De klant en product owner hebben geen enkele invloed op deze inschatting.

Maar hoe weet het team hoeveel uur ze aan een taak zal werken? En hoe worden die schattingen gemaakt? Hiervoor kunnen ze gebruik maken van een mechanisme wat “planning poker” wordt genoemd. Het team kan de inschattingen maken door gebruik te maken van werkuren of punten. Ze hebben dus een totaalbudget dat ze kunnen spenderen aan beschikbare taken. Het team zal dan door verbale communicatie te gebruiken beslissen hoeveel punten (of uren) ze aan elke taak zullen toekennen. Op die manier kunnen ze uiteindelijk beslissen hoeveel taken ze in totaal die sprint kunnen opnemen. Merk op dat deze planning poker opnieuw uitgevoerd wordt enkel door mensen die het werk zullen uitvoeren en niet door de mensen die het werk geven. De belangrijkste reden dat dit gebeurd is om “anchoring” te vermijden. Met anchoring bedoelen we dat het tam zijn keuze teveel kan laten beïnvloeden met gepresenteerde referentiepunten door externe partijen, zoals b.v. de product owner, waardoor hun objectiviteit wordt beïnvloed tijdens het inschatten.

2.3.4. Daily scrum

Dagelijks wordt er een stand-up meeting georganiseerd van 15 minuten waarbij het team de voortgang van de huidige sprint bepreekt. Tijdens dit overleg zoekt het team per lid een antwoord op de vragen: “Wat deed je gisteren?”, “Wat ga je vandaag doen?” en “Welke obstakels liggen in je weg?”. Het is belangrijk dat dit dagelijks gebeurd zodat indien nodig de planning van het team tijdens de sprint kan aangepast worden. Daarnaast moet dit best ook “live” gebeuren zodat het volledige team elke dag het volledige beeld ziet. Dit creëert ook sociale druk want iedereen moet zijn eigen vooruitgang vertellen en zeggen wat hij doet.

2.3.5. Sprint review & Retrospectieve

Op het einde van de sprint zal het team tijdens de sprint review een potentieel productiewaardig product aan de klant demonstreren, meestal door middel van een demo. Tijdens deze demo zijn ook het management en de product owners aanwezig. Het voordeel van een demo is dat de klant een product in werking kan zien en op die manier gericht feedback kan verschaffen over de applicatie.

Hoe weet het team of hun software productiewaardig is? Dit is een gevolg van de testen opgesteld tijdens de product planning. Er zijn echter verschillende definities voor te bepalen of software af is. Enerzijds kan je zeggen als er geen bugs meer in zitten, anderzijds kan je ook zeggen dat het ook effectief door de klant moet goedgekeurd zijn voor volledig af is.

Na de sprintdemo volgt er een intern evaluatiemoment, de “retrospectieve”. Tijdens dit bezinningsmoment waaraan het team, de scrummaster en optioneel de product owner deelnemen probeert men na te gaan wat er goed verlopen is tijdens de afgelopen sprint en wat er aan het proces kan verbeterd worden naar volgende sprints toe. We zoeken geen antwoord op de vraag “Waarom is taak X niet afgewerkt” omdat we dan na verloop van tijd telkens “te weinig budget” als antwoord zullen krijgen. Het doel is dus antwoorden te vinden over de werking van het proces op zich, niet over het inhoudelijke. Als resultaat van deze retrospectieve zullen er steeds een aantal acties bepaald worden om in de toekomst het proces optimaler te laten verlopen. Het kan ook zijn dat de product backlog moet aangepast worden maar dit mag enkel indien dit een meerwaarde heeft voor de klant.

2.3.6. Documenten

Scrum gebruikt heel weinig documenten. We hebben de product backlog, sprint backlog en burndown statistieken. De burndown statistieken worden enkel genomen van één sprint. Al deze documenten kunnen via een rekenblad opgesteld worden maar in realiteit worden meer automatische tools gebruikt zoals b.v. Trello of Jira.

In de burndown zien we grafisch de evolutie van het aantal resterende uren van een team gedurende de sprint. De grafiek bevat een neerwaartse lijn, vandaar de naam burndown, die het aantal resterende uren weergeeft die het team nog geeft om taken uit te voeren tijdens de sprint. Als deze te traag zakt dan is het tempo van het team veel te traag. Indien deze lijn voor het einde van de sprint 0 bereikt betekent dit dat het team te weinig taken heeft opgenomen want de sprint werd veel sneller afgewerkt dan voorzien.

3. KanBan

KanBan is een ontwikkelstrategie die vooral in productie-omgevingen / systeembeheer gebruikt wordt. Het gebruikt geen rollen zoals bij scrum wel het geval was maar focust vooral op het optimaliseren (lean) van de werking, doorvoer van ticketen, van het team. Er zijn geen sprints omdat het project beschouwt wordt als een continue proces waarbij we volgens het principe van “Just In Time” (JIT) werken. Taken/kaartjes worden dus aan het bord toegevoegd wanneer het probleem zich voordoet en moet aangepakt worden. Het is dus uitermate belangrijk dat deze manier van werken er voor zorgt dat er geen bottle necks kunnen optreden bij het uitvoeren van de taken. Als het team te veel taken tegelijk opneemt en deze niet tijdig kan afwerken dan zal de werking vastlopen. Het is dus belangrijk om een tempo te vinden dat het team aankan. Dit tempo wordt ook wel de WIP limiet genoemd of “Work In Progress” limiet. Het team mag dus niet meer actieve kaarten hebben dan het aankan. Als er toch een nieuwe taak moet aangemaakt worden maar deze WIP limiet is bereikt, dan voegen we het gewoon toe aan de product backlog zodat het op een later moment kan uitgevoerd worden.