

Gevorderd standalone

Plannen en ontwerpen

hoofdstuk

1

van Waterval naar Agile /
SCRUM





Algemene informatie

Onderwerp	Waterval en SCRUM/Agile
Leerdoel(en)	<p>De student leert de nieuwe methodiek (Scrum/Agile) en de oude methodiek (waterval) om IT projecten uit te voeren</p> <p>De student leert dat zijn kwalificatiedossier gebaseerd is op de Scrum/Agile methode/gedachtengoed</p>
Vereiste voorkennis	De student kent de faseringen in een ontwikkeltraject
Kwalificatiedossier	<ul style="list-style-type: none"><input checked="" type="checkbox"/> B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang<input checked="" type="checkbox"/> B1-K1-W2: Ontwerpt software<input type="checkbox"/> B1-K1-W3: Realiseert (onderdelen van) software<input type="checkbox"/> B1-K1-W4: Test software<input checked="" type="checkbox"/> B1-K1-W5: Doet verbetervoorstellen voor de software <input checked="" type="checkbox"/> B1-K2-W1: Voert overleg<input checked="" type="checkbox"/> B1-K2-W2: Presenteert het opgeleverde werk<input checked="" type="checkbox"/> B1-K2-W3: Reflecteert op het werk



Inhoudsopgave

Algemene informatie	2
Inhoudsopgave	3
Inhoud	4
1. Het examen/Proeve van Bekwaamheid (PVB)	4
2. Agile en waterval	4
3. Invulling van Agile/Scrum in DevOps	5
3.1 Wat is DevOps?	5
3.2 Product backlog	6
3.3 Sprint en sprint backlog	7
3.4 Board	7
4. Samenwerken in een Scrum team	9
4.1 Scrumteam	9
4.2 Verplichte en vaste overlegmomenten tijdens/voor een sprint	10
4.3 Pokersessie	12
5. Product backlog	13
5.1 Product backlog en sprints	15
5.2 Eigenschappen van een userstory (work item)	16
5.3 Userstory statussen	18
5.4 Userstories - taken	19
6. Overzicht oefeningen bij dit hoofdstuk	20



1. Het examen/Proeve van Bekwaamheid (PVB)

Het kwalificatiedossier voor een MBO tot software developer vereist voor het behalen van het diploma een Proeve van bekwaamheid (PVB) examen waarin de kandidaat bewijst het niveau te hebben verkregen van een startende Software developer.

Het examendossier vereist ook dat de kandidaat bepaalde (o.a. samenwerkings- en overleg-) vaardigheden laat zien in een klein teamverband.

De actuele ideeën en methoden met betrekking tot projectaanpak en samenwerking in het bedrijfsleven zijn heel verschillend. Daarom is er voor een projectmethodiek gekozen die veel voorkomt en ook past in de eisen van het examendossier. De naam van de manier van werken is Scrum/Agile en zal voor onze opleiding op deze manier tijdens je PVB (examen) worden geëxamineerd.

2. Agile en waterval

Agile is de theorie achter het werken in een SCRUM team. De Agile gedachte is gebaseerd op een aantal principes.

De waterval methodiek is in een vorig thema behandeld. Als je niet meer weet wat de waterval methodiek inhoudt, zoek dit dan nog op in een reader van het vorige thema of lees wat [wikipedia](#) hierover vertelt.

Agile is ook bedacht omdat de waterval manier o.a. geen rekening houdt met een sterk veranderende wereld en dus ook veranderende wensen en eisen van de klant. Waterval houdt zich te sterk vast aan de verschillende bouwfases die achter mekaar plaatsvinden met vaste mijlpalen (o.a. documenten zoals bijvoorbeeld een functioneel ontwerp of projectplan) die er moeten zijn.

Agile zet zich af tegen **Waterval** door de volgende principes

- Mensen en hun onderlinge interactie boven processen en hulpmiddelen
- Werkende software boven allesomvattende documentatie
- Samenwerking met de klant boven contractonderhandelingen
- Inspelen op verandering boven het volgen van een plan

Agile is ontstaan omdat veel IT projecten in voorgaande jaren mislukten doordat zij de rode onderdelen te belangrijk vonden.



3. Invulling van Agile/Scrum in DevOps

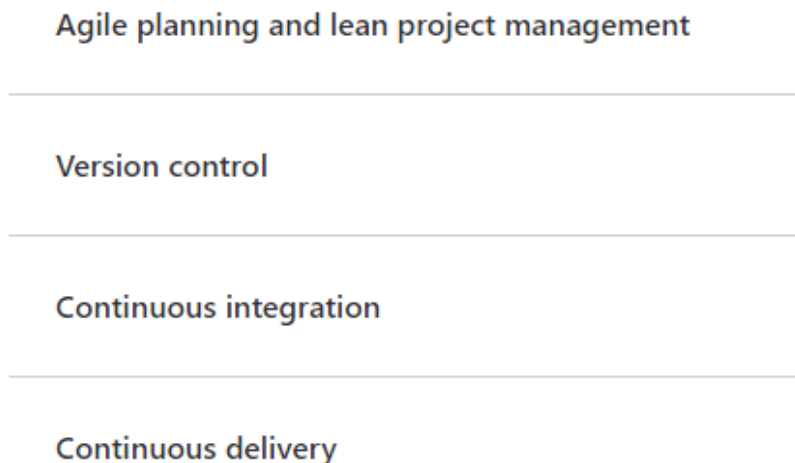
De DevOps omgeving van Microsoft biedt alle mogelijkheden om Agile/Scrum te werken en dat ook te oefenen. DevOps heeft echter een keuze gemaakt in hoe ze bepaalde zaken noemen en die wijkt soms af van wat er gebruikelijk in SCRUM qua gebruik van termen wordt gehanteerd.

Een aantal onderwerpen in deze reader zijn ook reeds behandeld in de readers van 'Testen en Verbeteren' en worden daarom ook minder uitvoerig behandeld.

3.1 Wat is DevOps?

DevOps is een manier van werken in softwareontwikkeling die de samenwerking tussen ontwikkeling en operations(=de IT afdeling die verantwoordelijk is voor het opleveren van nieuwe versies en het beheer van de applicaties) bevordert, wat resulteert in een snellere en betrouwbaardere levering van software. DevOps wordt vaak een cultuur genoemd en verbindt mensen, processen en technologie om continu waarde te leveren.

DevOps heeft o.a. de onderdelen:



Figuur 1 Hoofdonderdelen in een DevOps omgeving

Binnen deze reader wordt alleen de Agile planning/projectmanagement behandeld. Version control zal deels via Realiseren worden behandeld. Continuous integration en Continuous delivery valt buiten de scope van het kwalificatiedossier Software Development.

Om Agile planning en projectmanagement te kunnen leren heb je een allereerst een Devops omgeving nodig en moet je een project aanmaken.

Heb je nog geen Azure DevOps omgeving, creëer dan hier [gratis je eigen Azure DevOps](#) portal.



Agile project in DevOps

Heb je je eigen Azure DevOps portal maak dan eerst een project aan met het Agile template.

Create new project

Project name *
Agile project DevOps

Description
Project voor de reader Scrum/Agile

Visibility

Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

Advanced

Version control
Git

Work item process
Agile

Figuur 2, Nieuw project met Git en Agile

3.2 Product backlog

Product backlog bevat alle werkitems, die wanneer ze allemaal zijn afgerond, ook het volledig product klaar is.

Workitems bevat taken, die een werkitem verder in detail uitwerken en een taak kan worden toegewezen aan een teamlid.

Werkitems zijn (bijna) altijd **Userstories**. Userstories worden in het laatste deel van deze reader uitgelegd.

Hoe en met welke work items de product backlog wordt gevuld wordt in hoofdstuk 5 uitgelegd en gaan we daar ook mee oefenen.



3.3 Sprint en sprint backlog

Sprint of **iteratie** zijn termen voor hetzelfde begrip.

Een sprint is een periode van 1 of 2-4 weken waarin een deel van de functionaliteiten wordt aangepakt. **Een sprint moet een begin- en einddatum hebben.**

Iedere work-item/taak moet worden toegewezen aan een teamlid.

De functionaliteiten (is een deel van de product backlog) die aangepakt gaan worden is de sprint backlog.

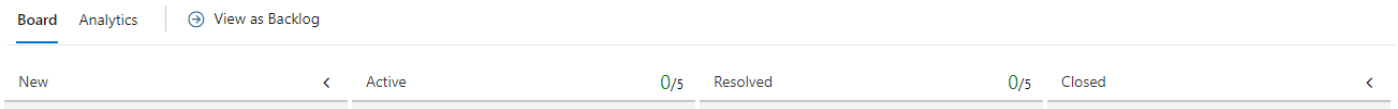
De sprint backlog wordt bepaald door het team natuurlijk wel in overleg met de klant of product owner.

3.4 Board

Het board is de visuele weergave van de stand van zaken van alle work-items of juist van 1 specifieke sprint of iteratie.

Er zijn veel varianten in de praktijk maar de variant hieronder zien we vaak terugkomen.

DevOps heeft zijn eigen variant en daar wordt dit een **board** genoemd en heeft het net andere kolom headers:



Figuur 3, SCRUM board in DevOps



De headers geven de status aan van een work-item.

New staat voor de Product Backlog

Active zijn de stories (story) (sprint backlog) die in de sprint moeten worden afgehandeld

Resolved zijn de workitems die volgens de ontwikkelaar afgehandeld/klaar zijn en kunnen worden gedemonstreerd aan de klant.

Closed zijn de workitems die ook getest zijn door een tester en door de klant in productie kunnen worden genomen.

Bij de module 'Testen en verbeteren' worden deze headers aangepast naar:

New => **To-Do**

Active => **In progress**

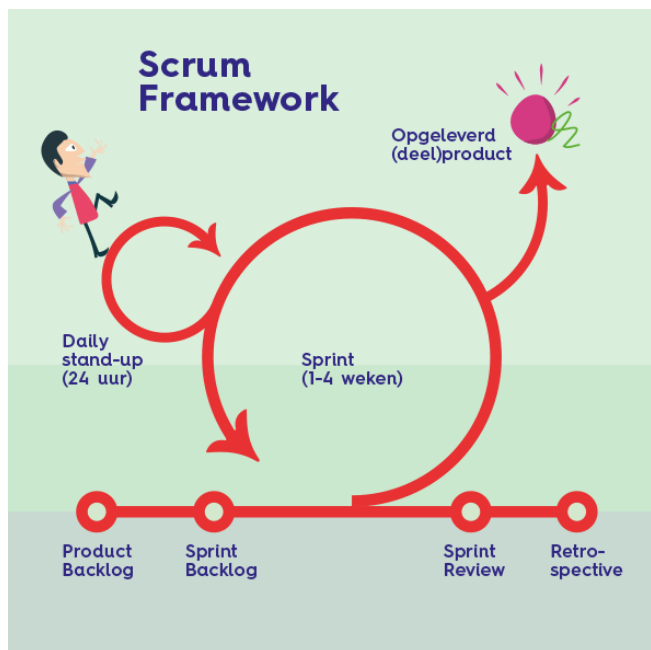
Resolved => **Testing**

Closed => **Done**

Bovenstaande nieuwe headers gaan we ook voor alle projecten hanteren omdat die meer overeenstemmen met de manier waarop we de verschillende statussen uitleggen en gebruiken.



4. Samenwerken in een Scrum team



Scrum beschrijft een concrete manier van werken in een iteratief (Agile) proces (sprint na sprint) in teamverband. In de praktijk is het team multidisciplinair, maar voor deze opleiding bevat het slechts je klasgenoten, je docent en tijdens de PVB een examinerator.

Voor Scrum wordt hier uitgelegd hoe een team is samengesteld en welke rol ieder in het team speelt. Daarnaast worden de typische verschillende overlegmomenten genoemd en besproken.

4.1 Scrumteam

Een scrumteam bestaat in de praktijk uit 3 tot 10 leden. Op school en in de examencontext zullen dit er nooit meer de dan 3 of 4 zijn.

Het team is samen verantwoordelijk. Als bepaalde taken worden uitgevoerd door iemand binnen het team die daar de benodigde vaardigheden voor heeft, is het team alsnog gezamenlijk verantwoordelijk voor het eindresultaat.

Ieder teamlid heeft een bepaalde rollen in het scrumteam en heeft daardoor een specifieke verantwoordelijkheid.

- Developer

Voert bijna alle workitems en taken uit

- Product owner

De Product Owner is verantwoordelijk voor het beheer van de Product Backlog. De product owner is de klant zelf of een collega die goed op hoogte is van de wensen van de klant en ook een directe lijn heeft met de klant.

De product owner bepaalt ook de prioriteit van een work item

De product owner kan ook een developer zijn. In de school/examen context zal de docent of examinerator deze rol gaan vervullen.

- Scrummaster

Taken van een scrummaster zijn:

- Zorgen dat de volledige sprint backlog aan het eind van de sprint klaar/afgerond is.



- Eventuele belemmeringen tijdens het project oplossen of uit de weg gaan.
- Zorgen dat iedereen begrijpt wat er gedaan moet worden en waarom.

In de praktijk is een scrummaster geen Developer, maar zal in de school context wel altijd het geval zijn zodat deze rol ook door leerlingen kan worden geoefend en hiermee ervaring kan worden opgedaan.

4.2 Verplichte en vaste overlegmomenten tijdens/voor een sprint

Vooraf, tijdens en na een sprint.



Sprint planning/ Poker session 1 overlegmoment met alle teamleden	Daily stand up Dagelijks aan het begin van de dag. Alle teamleden inclusief een scrummaster of een teamlid die de scrummaster rol op zich neemt.	Review Afsluitend overleg met alle teamleden en de klant of product owner	Retrospective Overleg met alleen de teamleden
--	--	---	---

- **Sprint planning**
Frequentie: 1 keer voor de start van een nieuwe sprint
Doel: Welke work items uit de product backlog wordt in de volgende sprint backlog geplaatst?
Hoe: Overleg met het team inclusief de product owner over welke work items een hogere prioriteit hebben

Resultaat:

Er is nieuwe sprint aangemaakt met een begin en einddatum, gevuld met alle userstories die in deze sprint worden opgepakt. Controle vindt plaats op de userstories of ze goed zijn omschreven, de acceptatie criteria puntsgewijs zijn benoemd, de storypoints en priority is toegekend. Ook moet de userstory aan 1 iemand worden toegewezen. Ieder teamlid zet de userstory die hij/zij als eerste gaat oppakken op ACTIVE.

- **Daily stand-up**
Frequentie: dagelijks
Doel: Een concrete status inzicht en waarna het scrumboard kan worden bijgewerkt
Hoe: Aan het begin van iedere werkdag vertelt iedereen de 3 volgende zaken: 1. Wat heb ik gisteren gedaan en wat is de status, 2. Wat ga ik vandaag doen, 3. Kan ik vooruit of heb



ik nog iets nodig? Ben ik ergens van of van iemand anders afhankelijk? Heb ik een belemmering (impediment) die de scrummaster voor hem/haar moet oplossen?

Resultaat:

De scrummaster of het teamlid dat de taak van scrummaster op zich heeft genomen zorgt ervoor dat de userstories en tasks naar de juiste status in het board worden gezet (in de juiste kolom (wordt vaak ook tijdens de daily-standup gedaan). De scrummaster gaat daarna ook meteen aan de slag om impediments uit de weg te helpen.

- **Review**

Frequentie: aan het einde van iedere sprint voor de Retro

Doel: Oplevering/demo van een volgende versie van een groeiend product aan de product owner/klant

Hoe: Het product wordt door een **Developer** gedemonstreerd en eventuele feedback wordt genoteerd of bediscussieerd.

Hier wordt ook bepaald of er nieuwe work items zijn ontstaan die in de product backlog moeten worden toegevoegd.

Resultaat:

Eventuele worden userstories aangepast en/of issues aangemaakt die in een volgende sprint worden meegenomen. De review en de retro worden uitvoeriger bij de module Testen & verbeteren behandeld.

- **Retro**

Frequentie: aan het einde van de sprint

Doel: Verbetering van het iteratieve proces, Verhoging van productiviteit en efficiëntie van het team, verbetering van de samenwerking.

Hoe: Developers met Scrummaster bespreken wat er goed en slecht (product, proces en samenwerking) is gegaan in de afgelopen sprint.

Resultaat:

De retrospective in DevOps wordt ingevuld. Alles wat goed liep en alle verbeteringen in de werkwijze of samenwerking worden genoteerd.



4.3 Pokersessie

Pokersessie of ook wel pokerplanning is een overleg techniek waarbij het hele ontwikkelteam betrokken is. De pokersessie is onderdeel van een sprintplanning en heeft als doel om een goede inschatting te maken van de complexiteit van de van de verschillende userstories en taken. Uiteindelijk is het de bedoeling dat ieder teamlid achter de keuze staat van het aantal storypoints omdat er bij verschil van mening over een taak een discussie ontstaat die wordt 'uitgevochten'.

De stappen in de pokersessie zijn als volgt:

1. Het team komt samen voor de sprint planning sessie en bespreekt de taken die in de sprint moeten worden uitgevoerd.
2. Een van de taken wordt geselecteerd en toegelicht door de product owner of de scrum master. Het is belangrijk dat iedereen het begrijpt en alle vragen zijn beantwoord.
3. Elke deelnemer krijgt een set kaarten met getallen, meestal gebaseerd op de Fibonacci-reeks, zoals 1, 2, 3, 5, 8, 13, 21, enzovoort.
4. Het team kiest een kaart die overeenkomt met hun inschatting van de complexiteit van de taak en legt deze kaart vervolgens met de achterkant naar boven op tafel. Het is belangrijk dat niemand anders de kaart kan zien.
5. Als iedereen een kaart heeft gekozen, wordt deze tegelijkertijd omgedraaid.
6. Als de schattingen van de teamleden overeenkomen, gaat het proces verder met de volgende taak. Als de schattingen sterk verschillen, leggen de teamleden uit waarom ze voor hun schatting hebben gekozen en kan er een discussie ontstaan over de details van de taak.
7. Na de discussie kiest iedereen opnieuw een kaart en worden deze tegelijkertijd omgedraaid.
8. Dit proces wordt herhaald totdat het team het eens is over de inschatting van de taak. Het kan zijn dat het nodig is om enkele kaarten opzij te leggen als niemand voor die waarde heeft gekozen.
9. Het team herhaalt dit proces voor alle taken die tijdens de sprint gepland moeten worden.
10. Zodra alle taken zijn geschat, is de sprint planning sessie voltooid en kan het team beginnen met het uitvoeren van de taken in de sprint.



Oefening 1.3 gaat hierover



5. Product backlog

De product backlog zijn alle workitems samen die het volledig product afdekken en alle functionaliteiten bevatten.

Een work item kan zijn:

Vanuit de Use Cases geïnventariseerd:

- Een userstory (= een functionaliteit = een subproces uit de use cases)

Vanuit de analyse en daarna het opstellen van de datadictionary en de ERD:

- Werkzaamheden om de database, de tabellen en de constraints te creëren en het te voorzien van een testset via inserts of imports.

Vanuit de Use Cases en de schetsen/wireframes van de User Interface:

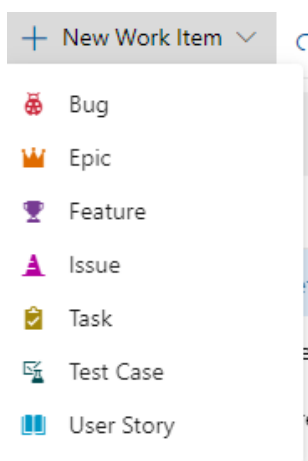
- Werkzaamheden om het beginscherm en de navigatie te implementeren

Vanuit een rollen en rechten onderzoek:

- Werkzaamheden om een login scherm te bouwen en ervoor te zorgen dat gebruikers een rol toegekend krijgen die bepaalt tot welke onderdelen zij wel/niet toegang krijgen.

De laatste soort work items (rollen en rechten) vallen voor onze opleiding buiten de scope, maar is wel een onderdeel die in de praktijk zeer zeker altijd van belang is.

Binnen de DevOps van Microsoft heb je de mogelijkheid om de volgende work items aan te maken:

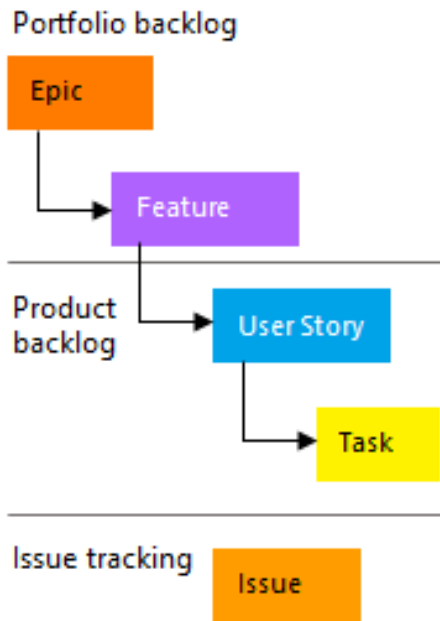


Tot nu toe zullen we deze allen userstories noemen. Een userstory die vanuit de klant / of vanuit één van de actoren is geschreven, heeft als **Area: Business**. De database acties en het beginscherm en de acties om de navigatie te implementeren hebben bij het definiëren van een work-tem de area: **Architectural**



Zoals de afbeelding hiernaast laat zien kunnen user story samengebundeld worden in een Feature, op kunnen meerdere Feature samen tot een Epic worden samengevat.

Omdat we geen grote en complexe applicaties ontwikkelen op het MBO vallen **Epic en feature buiten de scope** (en hoeven niet te worden gebruikt) van de opleiding.



Epic is een verzameling userstories die bij 1 bepaald onderdeel horen.

De term **Feature** gaan we in onze benadering **NIET** gebruiken.

Een **task** valt onder een work item. Als alle tasks zijn gedaan dan is een work item ook klaar.

Voor iedere **userstory** moet ook een **testcase** worden aangemaakt. Bij de module 'testen en verbeteren' wordt hier verder op ingegaan.

Bugs en Issues zal je bij het opstellen van de eerste product backlog niet gebruiken, omdat bugs pas ontstaan als er work items zijn geïmplementeerd en pas bij de eerste test case run kunnen worden gevonden. Bugs en issues slaan we in het kader van Plannen en ontwerpen over. Deze worden behandeld bij Testen en verbeteren.



Een omschrijving van de userstory die vaak wordt gebruikt is de volgende:

Een userstory wordt beschreven vanuit het gezichtspunt van een gebruiker, die de applicatie gebruikt. Iedere userstory die klaar is in de applicatie voegt iets toe waardoor de gebruiker in staat is meer waarde te creëren.

Na iedere sprint stijgt de waarde van de applicatie voor de gebruiker.

Als een user story niet geschreven is vanuit het perspectief van de gebruiker dan heeft het geen business value.

5.1 Product backlog en sprints

Voor de eerste sprint zal een sprint planningsoverleg bepalen welke work items uit de product backlog worden geïmplementeerd. De product owner zal hier een belangrijke stem hebben maar de mening van de Developer wordt zeker meegenomen.

Een sprint/iteratie zal minimaal een begin- en einddatum moeten hebben. De work items moeten aan het einde van een sprint klaar staan om aan de klant/product owner te kunnen worden gedemonstreerd in het review overleg.

Als de planning niet gehaald wordt zal de oorzaak worden besproken tijdens de retro zodat de volgende sprint wel een realistische planning wordt van aan te pakken userstories.



5.2 Eigenschappen van een userstory (work item)

Hieronder een afbeelding met alle eigenschappen die ingevuld moeten worden in het kader van jullie opleiding:

The screenshot shows a Jira user story form for 'US - 21 Toevoegen Klant'. Red handwritten annotations are present: '1' next to the title, '2' next to the assignee 'Abu Saebu', '3' next to the iteration 'Iteration 1', '4' next to the description 'Gebruiker kan een klant toevoegen', '5' next to the acceptance criteria, and '6', '7', and '8' next to the story points, priority, and value area respectively.

De genummerde onderdelen moeten ingevuld zijn:

1. De titel van de userstory Toevoegen Klant
2. Welke Developer gaat dit bouwen
3. Plaats de workitem in een Sprint/Iteratie
4. Een omschrijving/uitleg van de userstory heeft altijd als formaat:

De structuur van een userstory op Wie? Wat? en Waarom? van een userstory.

Voor een userstory met **Business** value:

Gedacht vanuit de klant

**Als <actor/persona/rol>
wil ik <beschrijving van wat ontwikkeld moet worden>
zodat ik <beschrijving van de reden/het doel waarom het ontwikkeld moet worden>**

Voorbeeld:

**Als beheerder van de applicatie
wil ik nieuwe gebruikers kunnen toevoegen
zodat nieuwe gebruikers in de applicatie kunnen inloggen**

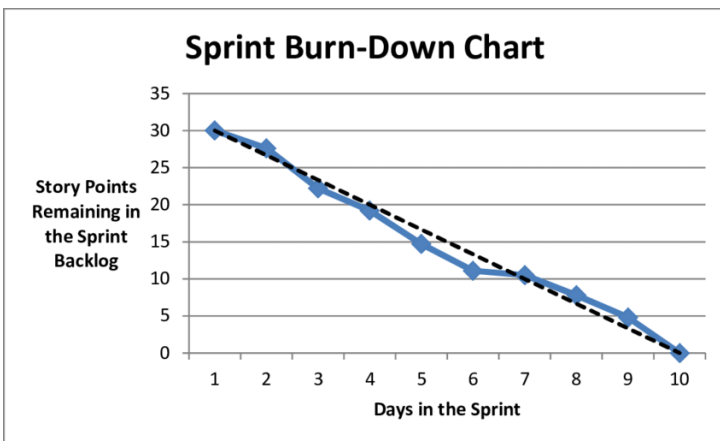
Hoe is de structuur van een userstory in de area **architectural**

Gedacht vanuit de ontwikkelaar:

**Als ontwikkelaar
wil ik een openingsscherm met menu
zodat ik alle menuonderdelen vanuit het beginscherm kan openen**




5. De acceptatie criteria
 - a. De acceptatie criteria zijn genummerd
 - b. Voor ieder criterium moet er een testcase op te stellen zijn
6. Story points: Geef een waarde uit de reeks (Fibonacci reeks) [1,2,3,5,8,13]. Geeft aan hoeveel tijd of hoe complex een userstory is om het volledig te bouwen zodat het aan alle acceptatiecriteria voldoet. (De waarde ervan wordt bepaald in een pokersessie vooraf aan iedere sprint.)
7. Priority – Hoe belangrijk is het voor de klant dat deze userstory in de applicatie werkt {1,2,3,4}. 1 is de hoogste prioriteit en 4 is de laagste prioriteit.
8. Value area: Business voor alle userstories die echt een toegevoegde waarde hebben voor de klant, Architectural voor de andere userstories (databases, testset en andere scripts, startscherm inc navigatie etc) zonder toegevoegde waarde voor de klant.



Door storypoints toe te kennen aan work-items kun je visueel weergeven door middel van een burndown chart of de werkzaamheden volledig volgens planning lopen.

De sprint bevat bijvoorbeeld in totaal 30 story points en de sprint duurt (in dit geval) 10 dagen. Na iedere daily stand-up overleg kan er een nieuwe punt worden gezet in de burndown chart.

 Importeren van Work-items in je DevOps board via een csv bestand? [Lees hier hoe je dit kan doen.](#)



5.3 Userstory statussen



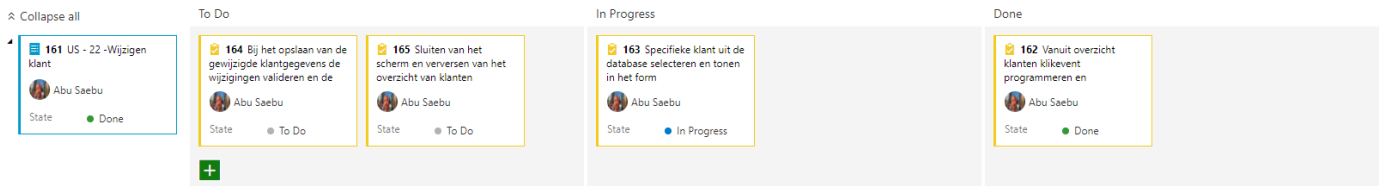
Figuur 4, Mogelijke statuswijzigingen voor een userstory work item



5.4 Userstories - taken

Een task of taak is een onderverdeling van één userstory. Een userstory is klaar wanneer alle onderliggende taken gedaan zijn.

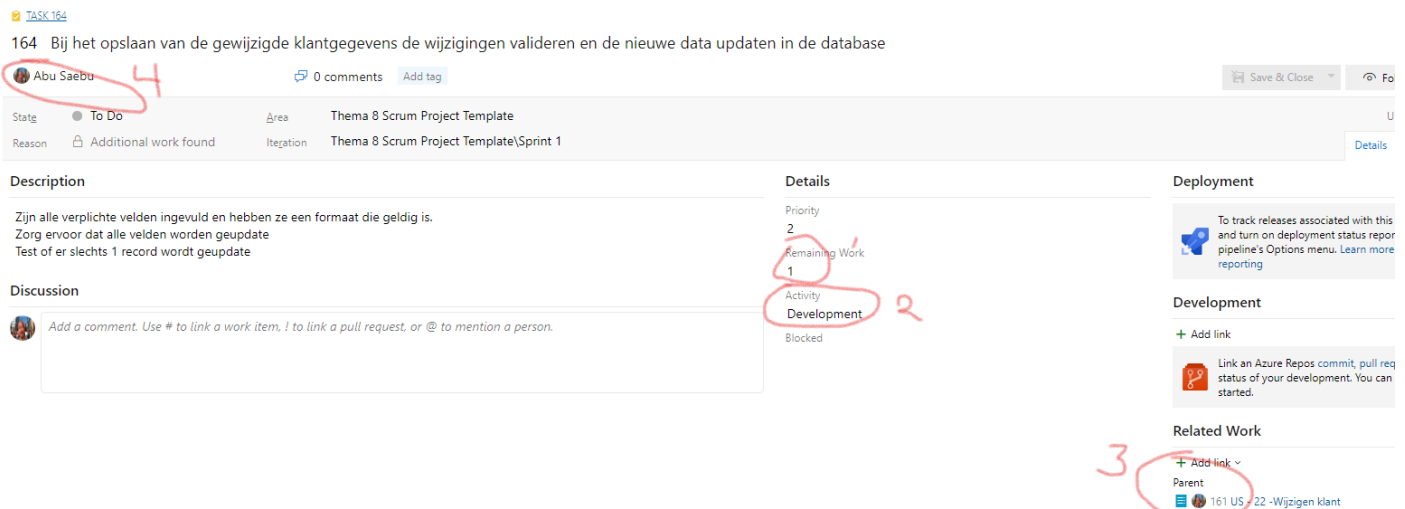
Taken zijn meestal technische werkzaamheden van wat er door een (back-end/front-end) ontwikkelaar/designer gedaan moet worden. De omschrijving mag ook technisch omschreven zijn.



Figuur 5, US 22 is opgedeeld in 4 programmeertaken

Iedere taak heeft als parent een userstory.

Als je een taak wijzigt kom je in het volgende formulier dat een beetje afwijkt van een userstory:



1. Geeft aan hoeveel werk te doen is voordat de taak klaar is
2. Activity = development voor een applicatie die nieuw gebouwd wordt
3. Parent is de Userstory
4. Een taak kan toegewezen zijn aan een ander teamlid



Een taak heeft geen **Acceptance criteria**. Die bestaan alleen op userstory niveau.

Bij het formulier van de userstory zie je nu dat er 4 children zijn:

Related Work

+ Add link ▾

Add an existing work item as a parent




Child (4)

- 164 Bij het opslaan van de gewijzigde klantgegevens de wijzig...
Updated just now, ● To Do
- 165 Sluiten van het scherm en verversen van het overzicht van ...
Updated just now, ● To Do
- 163 Specifieke klant uit de database selecteren en tonen in het...
Updated just now, ● In Progress
- 162 Vanuit overzicht klanten klikevent programmeren en wijzig...
Updated 2 minutes ago, ● Done

Als er met taken wordt gewerkt kan de voortgang veel preciezer worden ingeschat dan als er alleen met userstories wordt gewerkt.

Als alle taken van een userstory naar Done zijn gezet dan pas is de Userstory ook Done.

6. Overzicht oefeningen bij dit hoofdstuk

-  In **Oefening 1.1** oefen je het opstellen van userstories vanuit een case/context en zodoende de volledige product backlog te bepalen.
-  In **Oefening 1.2** oefen je het opdelen van de product backlog in sprints. Wat moet als eerste gedaan worden en wat kan pas later worden opgepakt?
-  In **Oefening 1.3** leer je hoe je als team de storypoints bepaalt in een pokersessie in een extension van DevOps die je vanuit de marketplace kan installeren.