Ontwerpen

Projecten in Scrum

Geschreven door Arjan Kamberg

https://www.linkedin.com/in/arjankamberg/

Leerjaar: 2022/ 2023

Ontwerpen projecten in Scrum
© 2023, Arjan Kamberg
Uitgegeven in eigen beheer

(Scrum@AKamberg.nl)

Versie Datum: 25-04-2023

Inhoudsopgave

Inleiding

In deze handleiding maken we kennis met het opzetten van een project in Scrum. We hebben in deze lessenserie geen tijd om een heel project te maken en af te ronden. Het voordeel van Scrum is dat dat ook niet hoeft. We gaan een omgeving inrichten waarvan uit steeds gekeken wordt wat we wel steeds kunnen afronden binnen de beschikbare tijd.

De hoofdstukken zijn zo ingericht dat ze per keer een logisch stukje van een scrum project beschrijven en we dat uitvoeren. Als we alle hoofdstukken hebben afgerond hebben we een Scrum project opgezet dat steeds een oplever product produceert per sprint.

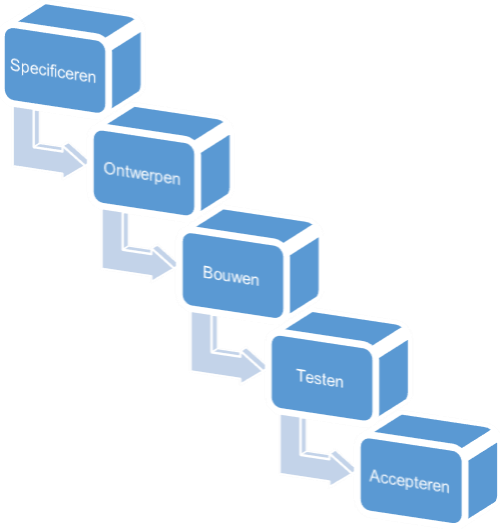
Voor het opzetten van een Scrum project hebben we geen voorkennis nodig. De lessenserie is gemaakt voor de opleiding Smart Technology, mbo-niveau 4 voor tweedejaars studenten. Omdat er wel daadwerkelijk iets gemaakt moet worden om ervaring op te doen met het gebruik wat geleerd is over scrum gaan we er in dit document vanuit dat de studenten kennis hebben van elektronische schakelingen gekoppeld aan een Arduino of een Raspberry pi. Daarnaast is er kennis van 3d ontwerpen en 3d printen voor het maken van een behuizing.

Deze lessen kunnen ook voor een ander vakgebied ingezet worden. De methoden die gebruikt worden zijn hetzelfde, alleen wat er gemaakt wordt zal anders zijn.

Wat is Scrum

Bij het uitvoeren van projecten in de softwarewereld ging er steeds iets mis. Projecten werden veel duurder dan begroot en ze werden veel te laat opgeleverd. Hoe kan dat.

Als je een huis wilt bouwen gebeurde dit niet. Van tevoren werd alles bedacht, materialen werden verzameld, fundering werd gemaakt en alle lagen erbovenop. Het maken van een huis gaat via een vast patroon dat we het *waterval* model noemen. Het waterval model staat voor een systeem waarbij de ene stap logisch volgt op de volgende.

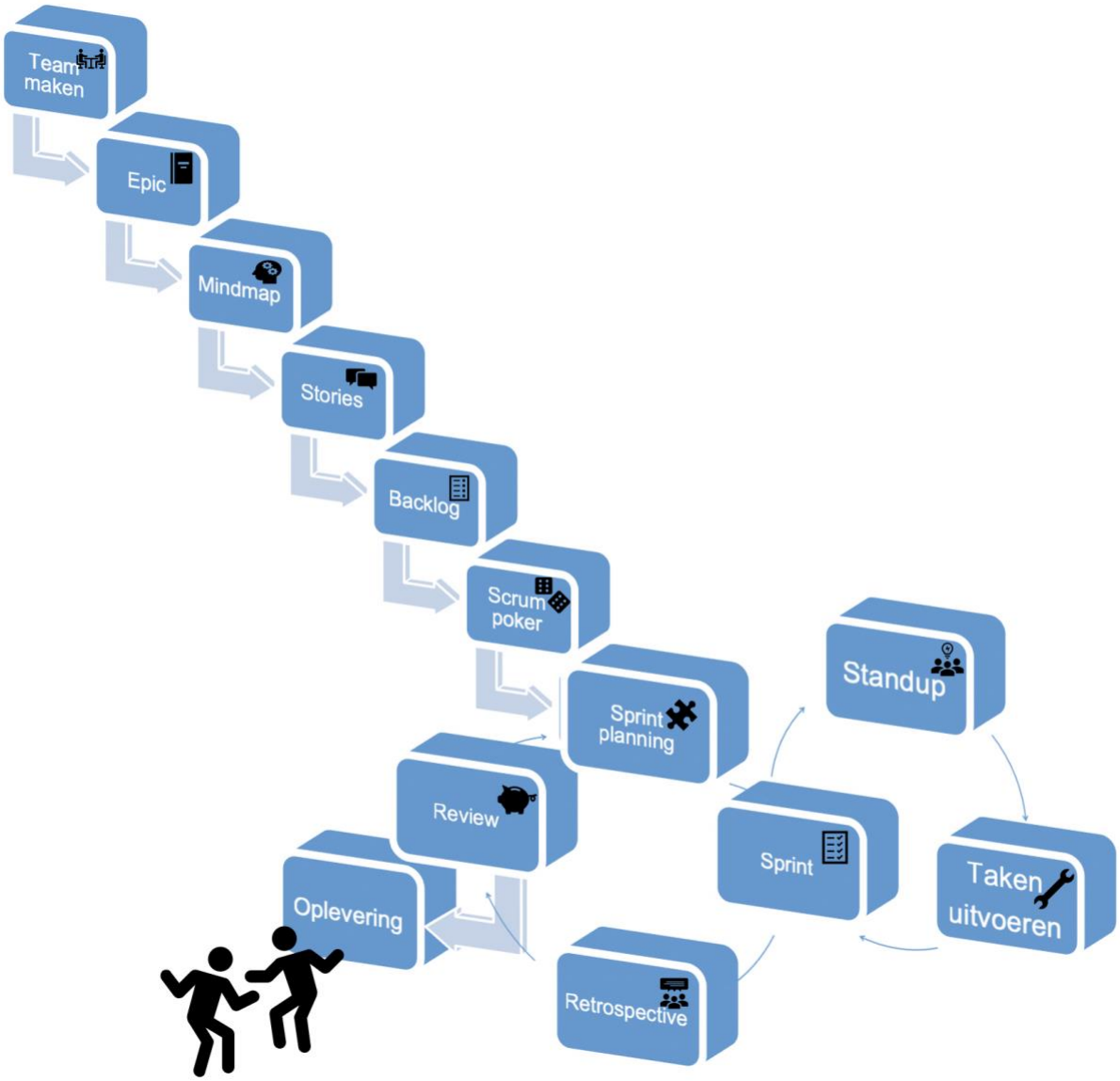


Software ontwikkelen is alleen anders dan het bouwen van een huis. Specificaties zijn vaak niet heel duidelijk van tevoren vastgelegd. Het ontwerp wordt dan wel gemaakt, maar tijdens het bouwen blijken er nieuwe eisen naar boven te komen. Als bij een huis de fundering gestort is kan daar niets meer aan veranderd worden, bij software is dat alleen anders, daar is op ieder moment nog iets te wijzigen. Daarbij gaat de technische vooruitgang erg snel, en komen er eisen tijdens het bouwen bij omdat er nieuwe apparaten ontwikkeld worden waar bij het ontwerpen niet over was nagedacht. Een voorbeeld is dat een oproepsysteem wordt ontwikkeld en na 5 jaar ontwikkelen kunnen mensen met een elektronische pieper opgeroepen worden. In de tussenliggende 5 jaar heeft iedereen een mobiele telefoon en niemand heeft meer een pieper. Het project wordt daarom niet geaccepteerd en er moet van alles veranderd worden om berichten naar de mobiele telefoons te sturen.

Een van de kenmerken van softwareontwikkeling is dat het eindresultaat vaak niet helemaal vast staat, dat de klant niet precies weet wat hij wil, en daarom is er ook geen eenduidige oplossing.

Als we het probleem vergelijken met het bouwen van een huis is het bouwen van software meer te vergelijken met het bouwen van een stad. Je weet dat je een stad wilt bouwen, maar je hebt nog geen idee hoe die er over 100 jaar uit zal zien. Je hebt wel ideeën over de richting waar je heen wilt maar de uitvoering doe je steeds per stapje. Na ieder blokhuizen ga je weer opnieuw nadenken hoe je verder wilt. Misschien haal je een blok huizen na 50 jaar weer weg om er iets beters neer te zetten en zo ga je steeds door. Dit proces is niet in een waterval methode te verwerken, maar wel in scrum.

In deze lessen serie gaan we een project opbouwen met scrum, en we leren de verschillende elementen van deze methode kennen.


Team maken
Epic
Mindmap
Stories
Backlog
Scrum poker
Sprint planning
Standup
Review
Sprint
Taken uitvoeren
Oplevering
Retrospective

Opzetten van een Scrum project

Voor het opzetten van een Scrum project wordt gebruik gemaakt van Microsoft Teams. Voor alle elementen die in scrum thuishoren is een programma gekozen dat in Microsoft Teams gebruikt kan worden.

Het Scrum team

In een scrum project zijn er 3 rollen die personen kunnen hebben. Dit zijn de *product owner*, de *scrum master* en de *team leden*.

Product owner

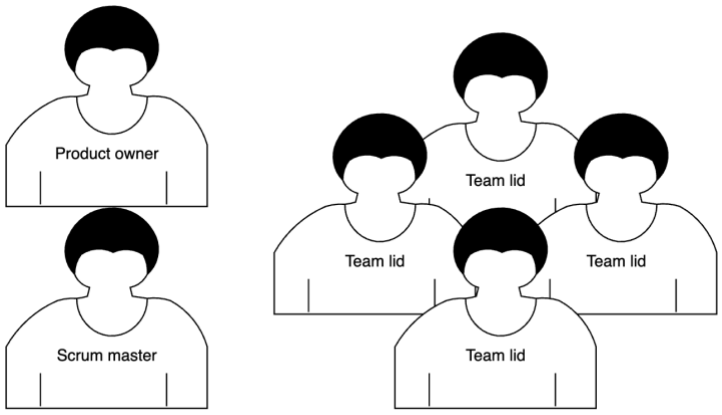
De *product owner* is eigenlijk de klant. In deze lessen is dit de docent die als klant speelt. De klant heeft eisen en de klant wil uiteindelijk een product hebben. Waar in het waterval model de klant helemaal aan het begin en helemaal aan het einde betrokken was bij het ontwikkelproces zal de klant nu veel vaker betrokken zijn. We gaan namelijk steeds kleine stapjes zetten in het project (de sprint), en na iedere sprint wordt aan de *product owner* getoond wat er de afgelopen tijd gedaan is. De *product owner* kan dan terugkoppelen of datgene dat gedaan is in lijn is met zijn verwachtingen. Of dat er nieuwe eisen of wensen naar boven komen na hetgeen dat hij gezien heeft.

Scrum master

De scrum master is een rol die ervoor zorgt dat alle stappen en taken netjes via de scrum methodiek gedaan wordt. Deze bereid de volgende sprint voor en houd de voortgang in de gaten. In dit project laten we de rol van scrum master een beetje los. De docent zal het proces in de gaten houden, en de voortgang. Het maken van een nieuwe sprint wordt nu door het gehele team gedaan.

Team leden

Een sprint team bestaat uit minimaal 4 personen, het liefst met verschillende kennisvlakken maar wel met een overlap van kennis. De 4 teamleden zijn allen gelijk aan elkaar, er is dus geen baas. Het idee is dat je met elkaar verantwoording neemt voor het project. Zelf plan je het project en zelf geef je sturing aan het project en zelf neem je dan ook verantwoordelijkheid.



Epic

In dit project gaan we anders te werk dan bij een project voor een bedrijf. We gaan zelf een product verzinnen dat we willen gaan maken. Omdat het hier gaat om een opleiding voor Smart Techonlogy moet de opdracht de volgende elementen hebben:

- Elektronische schakelingen
- Gekoppeld aan een Arduino of een Raspberry pi
- 3D behuizing ontwikkelen en 3d printer
- Communicatie via draadloze netwerken te besturen of uit te lezen op andere apparaten.

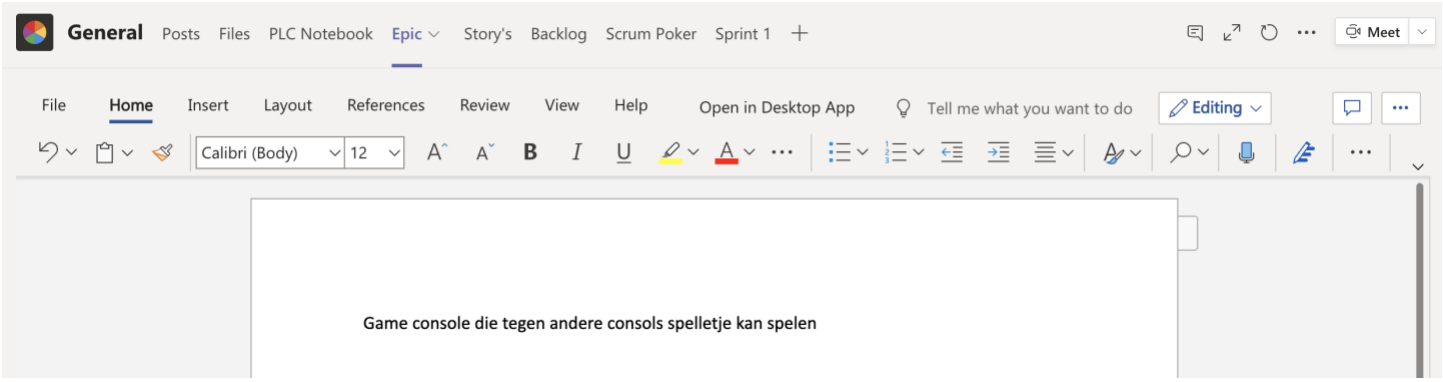
De docent zal bepalen of het project goed is, want die zal het als *product owner* dan als opdracht geven om het product te ontwikkelen.

We gaan eerst een *epic* maken, een *epic* is een korte beschrijving wat we gaan maken. Dit moet in een regel beschreven worden. We maken een *epic* om ons te focussen waar we mee bezig zijn. Een *epic* is wat je aan vrienden kan vertellen wat je aan het maken bent. Dus geen details maar een lekkere korte zin. Voorbeeld van een *epic* zijn :

- Slimme deurbel met camera.
- Lucht kwaliteit monitor met display.
- Auto verplaatsing detector met draadloze melder.
- Game console.

Dit zijn allemaal projecten waar gelijk vragen bij naar boven komen. Dat is niet erg. Als je dit aan je vrienden verteld en ze hebben vragen dan kunnen ze die stellen. Maar als ze geen vragen hebben weten ze gelijk in een zin zonder te veel details waar je mee bezig bent.

In teams zetten we bij *epic* het onderwerp die we hebben gekozen.



Opdracht : Verzin met je Team een Epic en zet deze in Teams.

Mindmap

Nu we het *Epic* hebben, hebben we ook gelijk heel veel ideeën waar we wat mee willen doen. Deze ideeën gaan we onderbrengen in een *mindmap*. Het gehele team maakt samen een *mindmap* met alles wat naar bovenkomt. Je hoeft niet nu te bedenken hoe je het gaat maken, of dat het realistisch is. Het gaat erom dat je nu creatief bezig bent en het is zonde dat je later dat leuke idee vergeet. Een scrum project behoord altijd een grote mate van plezier in zich te hebben, en leuke ideeën zorgen voor meer plezier in het uitvoeren.

Centraal in de *mindmap* zetten ze de *Epic*, misschien nog wat korter opgeschreven.

In onderstaand voorbeeld is een “turret” bedacht, met alles wat het team daarbij kon verzinnen. Misschien gaan ze niet alles maken, of krijgen ze later weer andere ideeën. Deze mindmap is ervoor om later weer op terug te kijken als we niet weten wat we nog kunnen doen. We kunnen aan het begin goede ideeën hebben die we later weer vergeten. Dit is de plek om dat vastgelegd te hebben. Tijdens het project mag je de mindmap natuurlijk ook nog uitbreiden. Haal eigenlijk niets weg want een stukje in de mindmap kan je misschien ook weer aansturen om wat anders te verzinnen.

Opdracht : Maak van je Epic de mindmap in Teams.

Stories

In Teams zien we ook een tabblad met de kop *stories*. De *stories* worden meestal door de *product owner* geschreven. De *product owner* weet welke wensen er zijn voor verschillende personen die te maken krijgen met het product. *Stories* zijn beschrijvingen waarom we iets willen. Een *story* heeft altijd een vaste vorm hoe het beschreven is.

Als <rol> wil ik <wens> omdat <reden>.

Alleen de rode onderdelen mogen we veranderen. Deze zinnen zijn ervoor dat we gaan verzinnen waarom we iets willen. Alles dat we willen heeft een reden, maar soms weten we die reden na een tijd niet meer. We kunnen ook in verschillende *rollen* andere wensen hebben.

- Data moeten digitaal beschikbaar zijn
- Data moeten consistent en up-to-date zijn
- Data moeten conform AVG beschermd én afgeschermd zijn
- Data formats moeten platform-onafhankelijk zijn

In een story beschrijf je dan het volgende:

Als Systeembeheerder wil ik dat de data digitaal beschikbaar moet zijn omdat er dan continue een back-up gemaakt kan worden.
Als klant wil ik dat de data AVG beschermd zijn omdat ik niet wil dat iedereen bij mijn gegevens kan.
Als bedrijf wil ik dat de data AVG beschermd zijn omdat ik schadeclaims kan krijgen als dat niet zo is.
Als data-analist wil ik dat de data consistent en up to date is omdat ik betrouwbare concilies moet geven.

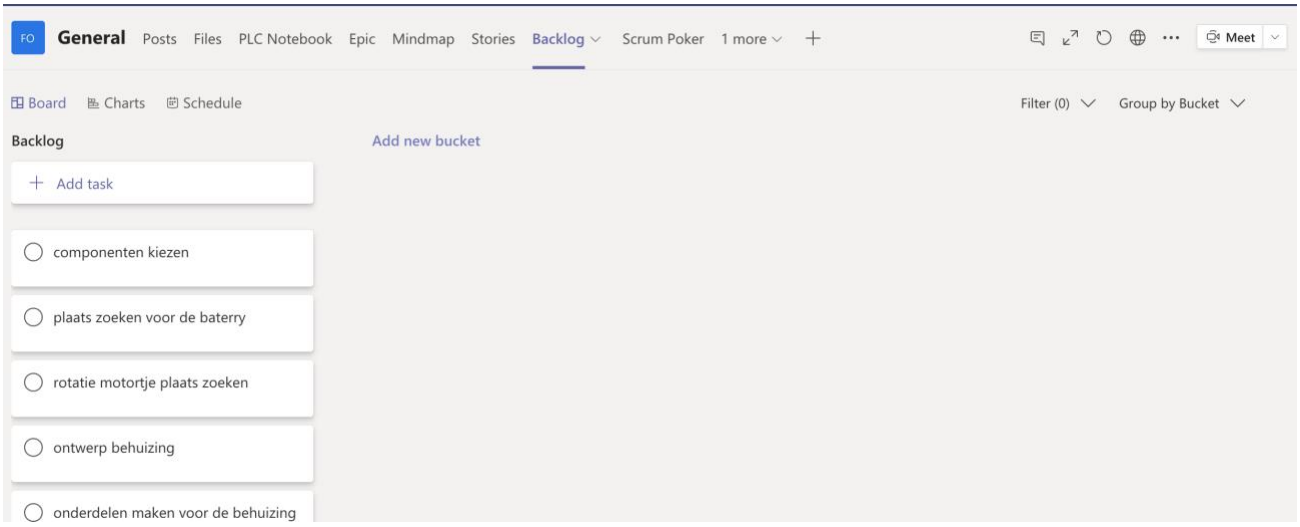
Opdracht: Schijf minstens 5 *stories* bij je project in Teams.

Backlog

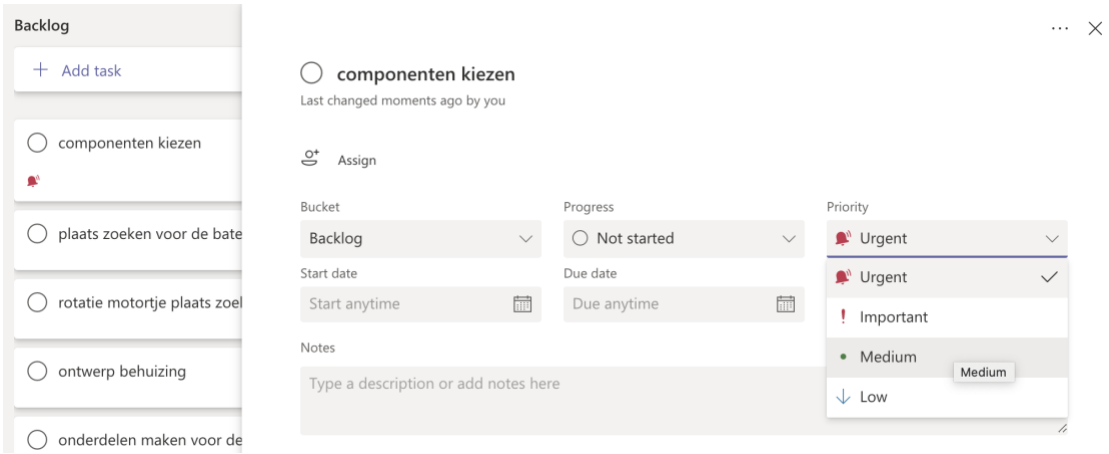
We zijn nu op het punt dat we dingen willen gaan maken. We hebben nog geen taken die we kunnen gaan maken, alleen nog maar ideeën en redenen. In de *backlog* gaan we echte taken beschrijven. Zoals een 3D ontwerp maken voor de behuizing van ons product. Het 3D printen van de behuizing. Het verzamelen van de datasheets van de elektronische onderdelen. Dit zijn dus echte talen die een teamlid kan gaan uitvoeren. De taken moeten goed omschreven worden, zijn ze dat niet dan komen we hier in het volgende hoofdstuk op terug.

We zien in Teams een kolom met de naam *backlog*. Daar zetten we alle taken die we nu kunnen verzinnen. Hebben we tijdens het uitvoeren van het project een nieuw goed idee dan kunnen we die als taak toevoegen hier. Iedereen mag extra taken toevoegen in de *backlog*.

De *backlog* kan er dan als volgt uitzien. Bedenk dat de *backlog* heel lang kan bestaan. Misschien wordt een taak die je nu hebt bedacht pas over een jaar echt gemaakt. Zorg er daarom voor dat de taak goed is beschreven.



Kom je tijdens het ontwikkelen later een error tegen dan is de *backlog* ook de plek om die error te melden. Heeft de error of de taak een hoge prioriteit dan kan dat aangegeven worden zoals hieronder staat.

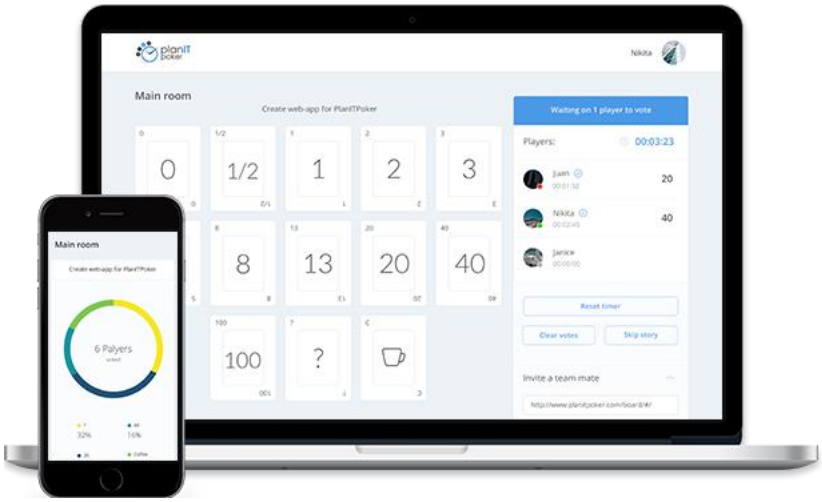


Opdracht : Verzin minstens 10 *taken* die in de *backlog* horen.

Scrum poker

In de *backlog* hebben we nu taken staan. Maar hoe lang duurt het voordat we een taak hebben uitgevoerd. Om dat te bepalen gaan we met het scrumteam pokeren. Dit pokeren doen we altijd zonder leidinggevenden of *product owners*. We willen als team een realistische planning afgeven zonder beïnvloed te worden door anderen. Zo is ook iedereen in het team gelijkwaardig en naar iedereen zijn meningen en ideeën moet geluisterd worden. Als we bij elkaar in dezelfde ruimte zitten is het leuk om echte Scrum-kaarten te gebruiken.

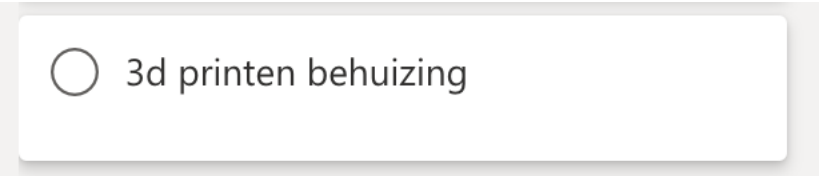
Als we die kaarten niet hebben zijn er ook apps die je kan gebruiken. Of je kan *Scrum poker* gebruiken dat in Microsoft Teams zit.



Het doel van het scrum poker is dat we door kaarten in te zetten een tijdsplanning gaan maken. De laagste kaart die we hebben is 0, dan komt 1/2 , 1, 2, 3 en dan opeens 5, 8, 13. Er zijn rare sprongen die we opeens maken. Voor het gemakt gaan we ervan uit dat 1 punt gelijk staat aan 1 uur. Dus 2 punten aan 2 uur.
Het is redelijk goed in te schatten of iets 1 uur of 2 uur duurt. Maar of iets 8 uur of 13 uur duurt is al een stuk lastiger. Hoe groter de tijd is waar we over schatten, hoe groter ook de onzekerheid zal worden.
De stappen tussen de kaarten hebben de logica van de Fibonacci-reeks. Iedere volgende kaart heeft de waarde van de 2 voorgaande kaarten.

Wat zijn de spelregels van het Scrum pokeren. We gaan een voor een alle taken langs in de backlog. Als er heel veel taken in de backlog staan dan beginnen we met degene met de hoogste urgentie.

We gaan nu uit van onderstaande taak:



Is deze taak duidelijk voor iedereen. Zo niet kunnen we nu de taak verduidelijken. Als dat gedaan is zet iedereen blind (dus dat de andere het niet kan zien) een kaart in met daarop de storypoints (uren) dat je denkt dat het duurt om deze taak uit te voeren.

Naam	Waarde
Teamlid 1	2
Teamlid 2	1
Teamlid 3	2
Teamlid 4	5

Als ieder teamlid heeft ingezet draaien we allemaal de kaarten om, en zien dan dat de meeste 1 a 2 uur denken en er is er 1 die 5 uur denkt. Die 5 uur is niet fout, want daar gaat het nu niet om. Die 5 uur is de opening van discussie tussen de teamleden. Degene die 1 uur heeft gezegd en degene die 5 uur heeft gezegd gaan aan elkaar vertellen waarom ze denken dat de taak respectievelijk 1 of 5 uur duurt. Teamlid 4 zegt dat hij nog nooit een 3d-printer heeft gezien. Hij weet niet of al het materiaal aanwezig is en of de printer van school werkt met het materiaal waar ze voor hebben gekozen. En welk materiaal hebben ze

gekozen, dat kan Teamlid 4 ook helemaal niet zien. Teamlid 2 verteld dat hij vandaag nog heeft geprint op school, en dat dat heel makkelijk ging. Er is op school maar 1 materiaal en dat werkt goed.

Het team kan nu de beschrijving van de taak aanpassen. Ze kunnen samen besluiten dat er 2 taken gemaakt worden, een voor het uitzoeken hoe de printer werkt en welk materiaal, en 1 voor het daadwerkelijk printen. Of ze kunnen erbij zetten dat ze het enige materiaal op school gebruiken.

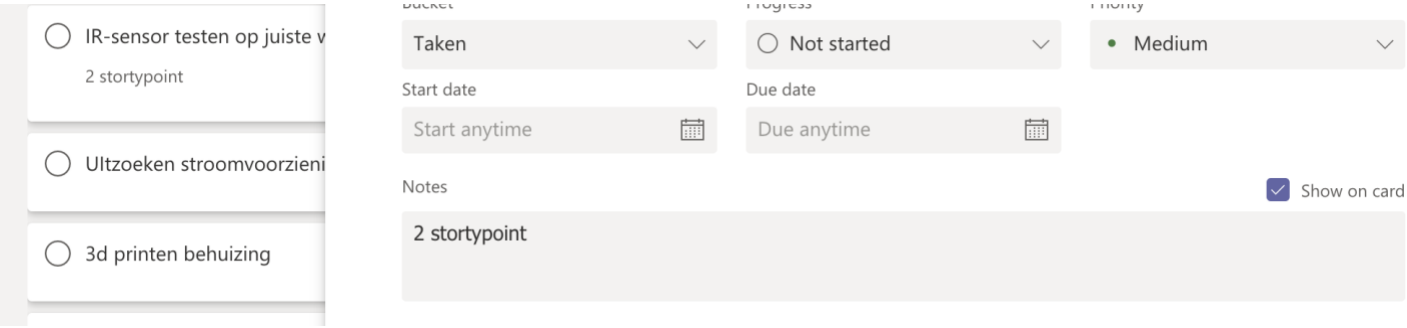
Nu gaat iedereen weer de kaarten oppakken en over de taak wordt weer gepokerd.

Naam	Waarde
Teamlid 1	1
Teamlid 2	2
Teamlid 3	2
Teamlid 4	2

Iedereen zet nu 2 in, want Teamlid 2 had er wel wat te makkelijk over gedacht, en teamlid 4 ziet nu dat het eigenlijk makkelijker is dan dat hij dacht. Het kan zijn dat er Teamlid 1 denk nu aan 1 uur. Dit ligt zo dicht bij elkaar dat het volledige team overeenkomt dat er 2 uur voor nodig is. We gaan geen gemiddelden nemen, we nemen een mooie ronde waarde.

Het kan ook zijn dat de teamleden nog steeds niet op dezelfde waarde uitkomt. Dan is het bijvoorbeeld goed om de taak beter te beschrijven, of op te splitsen in meerdere taken. Dit doen we net zo lang dat iedereen het erover eens is.

De *storypoints* worden bij de *Notes* ingevuld van de taak. Door de checkbox “Show on card” aan te vinken zie je van alle taken hoe lang daarvoor bedacht is.



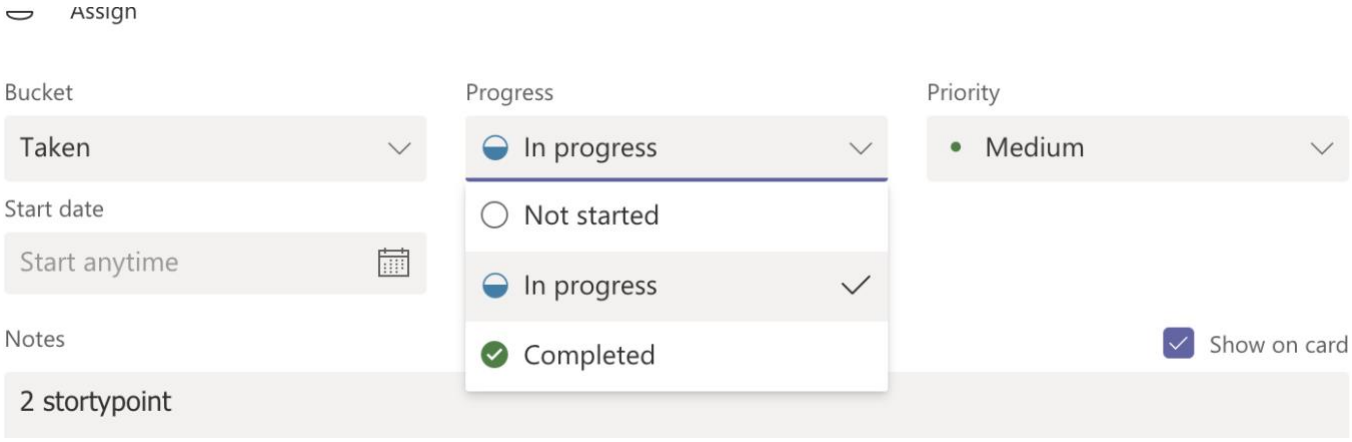
Opdracht: Alle taken in de *backlog* krijgen storypoints toegekend.

Sprint planning

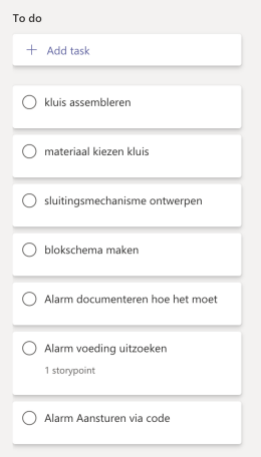
Over 3 weken wil de *product owner* zien hoe ver we zijn en of we op de goede weg zijn. We hebben een team van 4 leden, en per week kunnen we 1 uur aan taken werken.

Dat betekent dat we 3 x 4 x 1 = 12 uur aan het project kunnen werken de komende 3 weken. Anders gezegd kunnen we de komende 3 weken 12 *storypoints* doen.

We gaan nu met het team uit de *backlog* taken verzamelen die samen 12 storypoints zijn. Die taken zetten we op “In progress” en de taak kopiëren we ook naar de kolom met de naam “Sprint”.



De taken die we gekopieerd hebben gaan we nu splitsen in taken die allemaal 1 storypoint zijn, oftewel 1 uur duren. We hebben maar 1 uur per week de tijd om de taken uit te voeren, of deze manier kan ieder teamlid iedere week 1 taak uitvoeren. Aan het einde van de week kan je dan zien dat er 4 taken zijn afgerond.



Alle taken staan in eerste instantie in de “To do” lijst.

Als we deze taken hebben verplaatst dan kunnen we met de sprint beginnen.

Opdracht : In de kolom sprint staan 12 taken van 1 *storypoint*. (bij 4 teamleden).

Sprint en Standup

We hebben nu in de sprint een lijst van taken die we met elkaar hebben afgesproken dat we gaan uitvoeren. We kunnen nu geen andere taken gaan uitvoeren, want die staan niet geplant in de sprint.

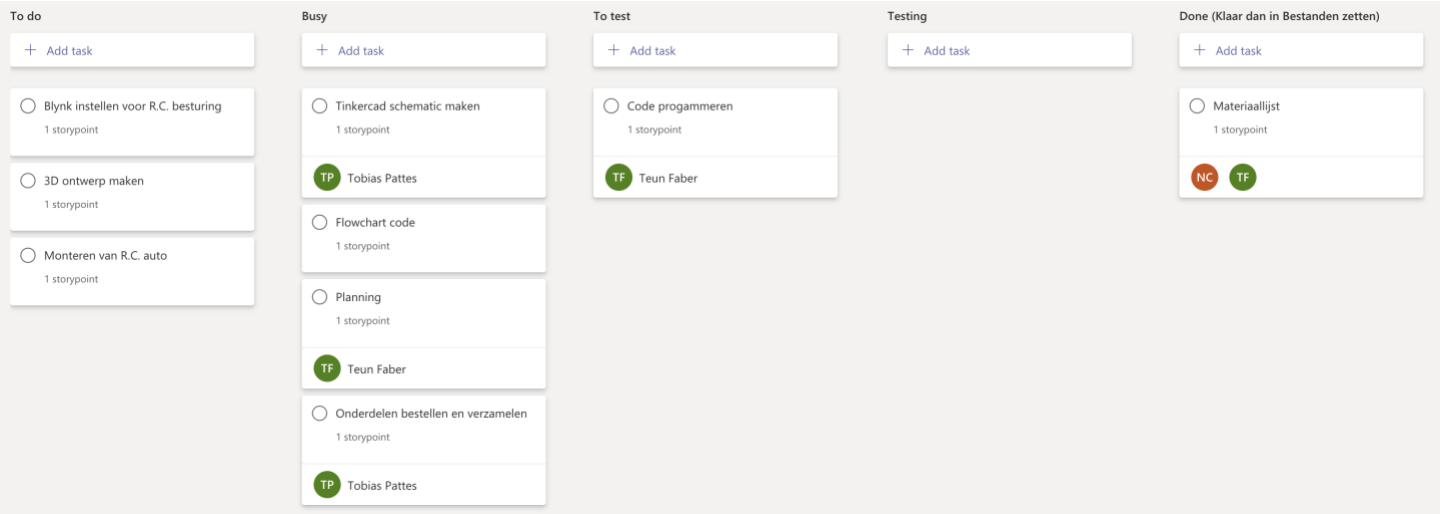
Aan het begin van de dag beginnen we de werkzaamheden met een *standup*. Een standup doen we zoals het woord al zegt staande. De reden dat dit staande gebeurd is dat staan niet zo comfortabel is. Als we gaan zitten is de verleiding sterk om ook op je praatstoel terecht te komen. Een standup duurt maximaal 5 a 10 minuten. De sprint master houdt in de gaten dat ieder zijn verhaal kort is. Het is ook de bedoeling dat we allemaal goed naar elkaar luisteren, zodat je weet wat een ander heeft verteld. Als we de standup digitaal doen is het lastiger om echt te gaan staan. We moeten wel in de gaten houden dat de verhalen niet te lang gaan duren.

Voor de standup kan je kijken welke taak je zou willen doen. Het is handig om ook te bedenken of de taak al gedaan kan worden, sommige taken kan je pas uitvoeren als er voorwerk gedaan is. Dat voorwerk moet per definitie ook een taak zijn (geweest).

Er zijn 4 vragen die iedereen om de beurt gaat vertellen:

1. Wat heb je de vorige keer gedaan.
2. Wat heb je geleerd of liep je tegenaan.
3. Wat ga je vandaag doen
4. Heb je daarbij hulp nodig.

Als dit klaar is gaan we aan de slag om de taken uit te gaan voeren. Iedereen gaat maar 1 taak tegelijk uitvoeren. De taken staan op een *storyboard.* Bij sommige projecten is dit een echt bord met papiertjes erop. Wij gebruiken dus een elektronisch bord waar we overal vandaan ermee kunnen werken.



De taak die we gaan doen schuiven we van “To do” naar “Busy”, en we zetten onze naam erbij. Als we de taak afhebben dan schuiven we de taak een stap verder naar “To test”. Iemand anders moet nu controleren of je alles goed hebt uitgevoerd. Die zet ook zijn naam erbij en schuif de taak naar “Testing”. En staan bijvoorbeeld de bestanden in de map bestanden zodat iemand anders er later ook bij kan. Als alles in orde is kan de taak doorgeschoven worden naar “Done”.

Indien het niet in orde is wordt dat bij de taak geschreven en wordt de taak weer naar “To do” gesleept. Het is niet de bedoeling dat de tester dit gaat oplossen, of in ieder geval niet op dit moment.

Stel dat de tester iets tegenkomt dat heel veel extra tijd gaat kosten, en dus niet binnen de sprint valt, dan zal een nieuwe taak aangemaakt moeten worden, en deze wordt in de *backlog* gezet. De taak is dan wel klaar maar heeft een vervolgtaak gekregen.

Het kan ook zijn dat tijdens het maken een heel urgent probleem naar boven komt. Iets dat gelijk gedaan moet worden. Hier wordt dan eerst een taak voor aangemaakt, en er wordt een hoeveelheid storypoints aan toegekend. Met de sprint master wordt besproken of dit echt nu gedaan moet worden. Is dat zo dan wordt deze taak in de spint geplaatst. Maar als er een nieuwe taak in de sprint wordt geplaatst, moet er een andere taak uit de sprint gehaald worden. Dit is ook de taak van de sprint master. We kunnen namelijk niet meer storypoints in een sprint uitvoeren dan dat er tijd is.

Opdracht 1: Voer iedere dag de standup met je groep uit en neem deze standup op in teams. Op deze manier kan je later terugkijken of de standup helemaal goed is verlopen.

Opdracht 2: Ga de taken ook daadwerkelijk uitvoeren. Kijk kritisch of je de taken wel kan uitvoeren en of je goed hebt gedacht van tevoren of je wel alles kan uitvoeren wat in de taak staat. Is de taak duidelijk of zou dit in het vervolg beter kunnen.

Retrospective

Als de sprint ten einde is nemen we tijd met elkaar om te bespreken hoe de sprint is verlopen. Ging het goed met het verdelen van de taken. Was het leuk wat je hebt gedaan, en heb je wat geleerd. Dit gesprek is met de teamleden onderling. Iedereen is gelijk, en iedereen heeft respect voor elkaar. Dit is het moment om elkaar te verbeteren zodat de volgende sprint nog beter gaat worden. Een sprint team behoord ook geen leidinggevenden te hebben, en vooral niet in dit gesprek. Dit gesprek is open en eerlijk en iedereen moet dat ook voelen. Voor de retrospective mag je wel 15 minuten uittrekken, of zolang als nodig is als er wrijvingen in de groep weggewerkt moeten worden.

Opdracht : Aan het einde van de sprint voeren we een retrospective uit met je team.

Sprint evaluatie

Als de sprint afgelopen is wordt het behaalde resultaat aan de *product owner* getoond. Dit kan door de sprint master gebeuren, of door het hele team. De *product owner* geeft zijn visie op wat het ziet. Tijdens de demo komen er misschien nieuwe eisen naar boven of nieuwe ideeën. Na dit gesprek kan weer de volgende sprint ingeplant worden. Dit gaat door totdat het product opgeleverd kan worden.

Opdracht : Met het gehele team wordt de sprint evaluatie gedaan. Hiervoor nodig je de *product owner* uit (docent) waar het resultaat van de sprint aan getoond wordt.

Betere taken maken, Definition of done

Om betere tijdsinschattingen te maken voor een taak hebben we al gemerkt dat de taak heel goed beschreven moet worden. Hoe beter de beschrijving, hoe minder er later de uitvoering ter discussie komt te staan. Als we dit nog verder uitwerken komen we op het punt dat je de “Definition of done” beschrijven. Wanneer is een taak klaar. Dit is voor iedere taak en ieder soort taak steeds anders.

Het development team bepaal wat deze definitie is voor iedere taak. Voor een stuk software kan een definition of done betekenen dat de code succesvol door een serie unit tests heen gaat. Voor het uitzoeken van datasheets is de definition of done dat alle datasheets zijn opgeslagen in de folder “Datasheets” in het Teams files folder.

Oplevering

Aan het einde van een of meerdere sprints kan het zover zijn dat het product ook echt wordt opgeleverd. Dit hoeft niet het einde van de ontwikkeling van het product te zijn. Misschien gaan we wel verder als we nog meer *backlog* items hebben, en de *product owner* wil ook dat er verder ontwikkeld wordt. Dit moet wel een feestelijke mijlpaal zijn. Een moment om trots het product officieel te overhandigen.
Het kan ook zijn dat dit het einde is van een ontwikkeling. Misschien wordt het team wel gewijzigd of gaat het een ander product ontwikkelen.

De oplevering komt altijd na een sprint evaluatie. Het kan dus nooit zo zijn dat de *product owner* nu nog voor verrassingen komt te staan.

Opdracht: Als het product zover is dat deze opgeleverd kan worden dan wordt dit feestelijk gedaan. Er wordt een demo gegeven aan de *product owner*.