

scrum@work

De scrumgids

De gids voor Scrum, de regels van het spel

Deze scrumgids is een uitgave van scrum@work
www.scrumatwork.nl | info@scrumatwork.nl | 085 – 864 02 40

Inhoudsopgave

Aanleiding	3
Agile	3
Relatie Agile en Scrum	5
Scrum	5
Planning poker	10
Stakeholders	12
Scrum in de praktijk met Jip en Janneke	13
Tot slot	14
Bijlage 1: begrippenlijst	15
<i>4 rollen</i>	<i>15</i>
Product Owner	15
Scrummaster	15
ProjectTeam	15
Stakeholder	15
<i>4 meetings</i>	<i>15</i>
Planningsessie	15
Daily Scrum	15
Demo	16
Retrospective	16
<i>3 hulpmiddelen</i>	<i>16</i>
Product Backlog	16
Sprint Backlog	16
Acceptatie criteria	16
<i>Diversen</i>	<i>17</i>
User Story	17
Story point	17
Planning poker	17
Sprint	17
Sprintdoel	17
Bijlage 2: Scrum binnen en buiten IT	18

Aanleiding

Voordat u begint met lezen van deze scrumgids willen wij u vragen of u één van de volgende situaties herkent?

- Project loopt stroperig.
- Het project komt maar niet af.
- De klant is niet tevreden en de opdrachtgever wordt boos, omdat niet geleverd wordt wat gevraagd is.
- Het is niet duidelijk wanneer iets klaar is.
- De partijen die invloed hebben op een project worden niet of nauwelijks betrokken, geen commitment.
- Iedereen vindt er maar wat van maar er wordt alleen naar de hardste schreeuwer geluisterd.

Als u geen van deze situaties herkent mag u zich gelukkig prijzen en wat ons betreft stoppen met lezen. Dit boekje is bedoeld voor alle andere mensen die wel één of meerdere vragen met “ja” hebben beantwoord.

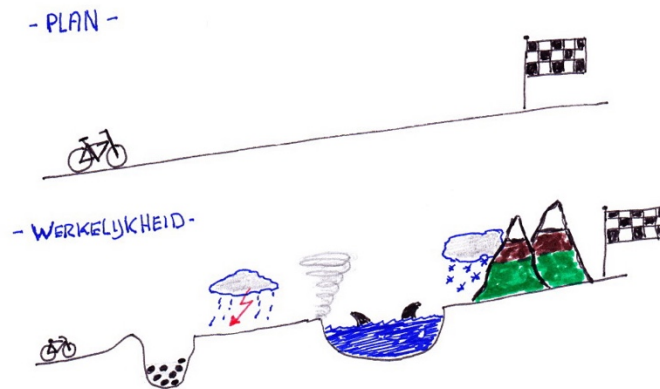
Dit boekje gaat over een andere manier van werken om de voornoemde projectteleurstellingen te voorkomen. Hoe kunt u anders werken, met meer effectiviteit zodat er sneller resultaat gehaald worden zonder dat dit ten koste van de controle gaat? Daar gaat deze scrumgids over. Deze scrumgids gaat over een nieuwe manier van werken, een manier van werken die beter voorbereid is op veranderingen tijdens het traject, een manier van werken waarbij er alleen maar werk gedaan wordt dat direct waarde oplevert. Dit boekje gaat over werken met Agile/Scrum.

Scrum is een projectaanpak die oorspronkelijk is bedacht door software ontwikkelaars in de IT (Informatie Technologie) sector. In de IT wordt de methode op grote schaal succesvol toegepast. Denk aan IT teams van bijvoorbeeld banken. Deze aanpak biedt juist ook kansen aan niet IT organisaties. Het is een aanpak die generiek toepasbaar is en grote voordelen biedt ten opzichte van de traditionele aanpakken. Er is een originele Scrumgids beschikbaar waarin de spelregels van Scrum worden uitgelegd. Deze Scrumgids is geschreven voor software ontwikkelaars binnen de IT. Er zijn grote verschillen tussen IT-projecten en niet IT-projecten (zie bijlage 2) waardoor de originele regels zoals opgenomen in de scrumgids niet één-op-één zijn toe te passen. Scrum is bedacht door programmeurs en de termen die ze erbij toepassen zijn voor mensen buiten de IT wereld soms nogal abstract. Wij hebben daarom de Scrumgids herschreven speciaal voor lezers en geïnteresseerden werkzaam buiten de IT. In onze aanpak gebruiken we alleen die onderdelen van Scrum die helpen en hebben we overbodige onderdelen weggelaten. Waar nodig hebben we de termen en teksten herschreven zodat de Scrumgids leesbaar wordt en beter aansluit bij de praktijk van een bredere doelgroep.

Agile

De wereld om ons heen verandert steeds sneller en dat vraagt om een flexibele projectaanpak die zich beter kan aanpassen aan de nieuwe omstandigheden dan traditioneel projectmanagement. Bij traditioneel projectmanagement, denk hierbij bijvoorbeeld aan Prince2, treft u woorden aan als projectplan, beheersing risico's, verantwoording, vaste processen, change proces maar ook een set aan uitgebreide documentatie past bij traditioneel projectmanagement. De organisatiestructuur is vooral hiërarchisch ingericht met een vooraf bepaalde overleg en governance structuur.

Traditioneel projectmanagement gaat er bij de start van het project van uit dat het project via een vlakke en voorspelbare weg zal verlopen. De praktijk is echter weerbarstiger. In plaats van een voorspelbaar traject, verlopen projecten vaak een stuk onvoorspelbaarder. Dat komt omdat zich tussen het moment van starten en afronden van een project heel veel wijzigingen hebben voorgedaan. Dat kunnen tegenvallers zijn in planningen, werk dat langer duurt dan gedacht maar ook gewijzigde klantwensen zoals nieuwe ideeën of inzichten.

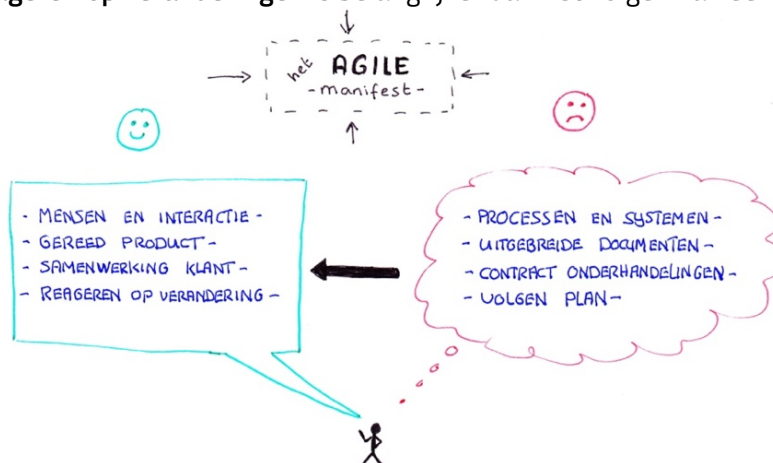


figuur 1: plan versus werkelijkheid

De flexibele wereld vraagt dus om een flexibele managementaanpak die voorbereid is op dit grillige projectpad. En daar biedt de Agile projectaanpak handvaten voor. Het Engelse woord Agile betekent "behendig", "lenig". De Agile projectaanpak is dan ook een aanpak waarin behendigheid in de aanpak voorop staat. De Agile projectaanpak probeert de veranderingen om ons heen zo goed mogelijk te faciliteren zonder daarbij het projectresultaat uit het oog te verliezen.

De Agile projectaanpak is gebaseerd op de uitgangspunten die in het Agile Manifest zijn vastgelegd. Het Agile Manifest is een internationaal manifest waarin de uitgangspunten van Agile staan beschreven:

- mensen en hun onderlinge interactie** is belangrijker dan processen en systemen
- een gereed product** is belangrijker dan uitgebreide documenten,
- samenwerking met de klant** is belangrijker dan contractonderhandelingen
- reageren op veranderingen** is belangrijker dan het volgen van een plan



figuur 2: het Agile manifest

Dit manifest bevat duidelijke uitgangspunten voor het succesvol toepassen van Agile. Naast dit manifest is de Agile projectaanpak gebaseerd op de volgende 12 principes:

- 1. Vroeg waardevolle producten opleveren.**
2. Frequent producten opleveren.
- 3. Wijzigingen zijn welkom.**
- 4. Opdrachtgever en team werken samen.**
5. Team met gemotiveerde mensen.
6. Face-to-face communicatie.
7. Voortgang meten aan werkende producten.
- 8. Werkdruk door het hele project gelijk.**
- 9. Continu aandacht voor verbeteringen.**
10. Hou het simpel, laat alles weg wat niet echt nodig is.
11. Zelf organiserende teams.
12. Kijk terug, leer, verbeter en ga verder.

Door deze uitgangspunten en principes toe te passen, wordt er een werkomgeving gecreëerd waarin teams kunnen accelereren, steeds beter worden en het project continue waardevolle producten oplevert.

Daar waar traditioneel projectmanagement vooral uitgaat van planmatig werken en de voortgang controleert t.o.v. dit plan (=plan gedreven), gaat Agile juist uit van het toevoegen van waarde (=waarde gedreven). Waar traditioneel projectmanagement zich vooral richt op de sturing van projecten richt Agile zich op het maak-proces van de producten binnen het project.

Relatie Agile en Scrum

Agile is een mindset, de manier van denken hoe tegen projecten en de uitvoering daarvan aan te kijken. De uitgangspunten en principes staan in het Agile Manifest. Scrum is een projectaanpak gebaseerd op het Agile Manifest en de Agile principes. Waar Agile gaat om de mindset, gaat Scrum echt om het doen, het in teamverband maken van producten, het uitvoeren van projecten, het effectief organiseren van werk. Samengevat: Agile denken en Scrum doen.

Agile/Scrum vormt door de combinatie van de mindset en principes uit het manifest én het effectief organiseren van werk, een projectaanpak die het mogelijk maakt om de complexiteit van een project te vereenvoudigen en de sturing te verbeteren.

Agile/Scrum typeert zich door begrippen als waarde toevoegend, wendbaarheid, lerend vermogen, vertrouwen, zelfsturende teams en flow. Dit alles tezamen zorgt voor verhoging van het werkplezier!

Scrum

De term Scrum komt uit de rugby en werd voor het eerst gebruikt door wetenschappers Ikujiro Nonaka en Hirotaka Takeuchi. Zij deden in de jaren tachtig onderzoek naar teamwerk en concludeerden dat projecten die worden uitgevoerd door kleine, multidisciplinaire teams historisch gezien het beste resultaat leveren.

Scrum is een projectmethode om in teamverband complexe producten in een complexe omgeving te maken. De nadruk ligt bij Scrum op het doen, het maken, het realiseren van producten door een team en biedt handvaten om dit zo effectief mogelijk voor elkaar te krijgen.

Het werken met Agile/Scrum als projectmethodiek leidt tot:

- kortere doorlooptijd van het project
- meer en actievere samenwerking
- duidelijke keuzes in prioriteit

Het raamwerk van de scrummethode bestaat uit:

4 rollen

1. Product Owner
2. Projectteam
3. Scrummaster
4. Stakeholder

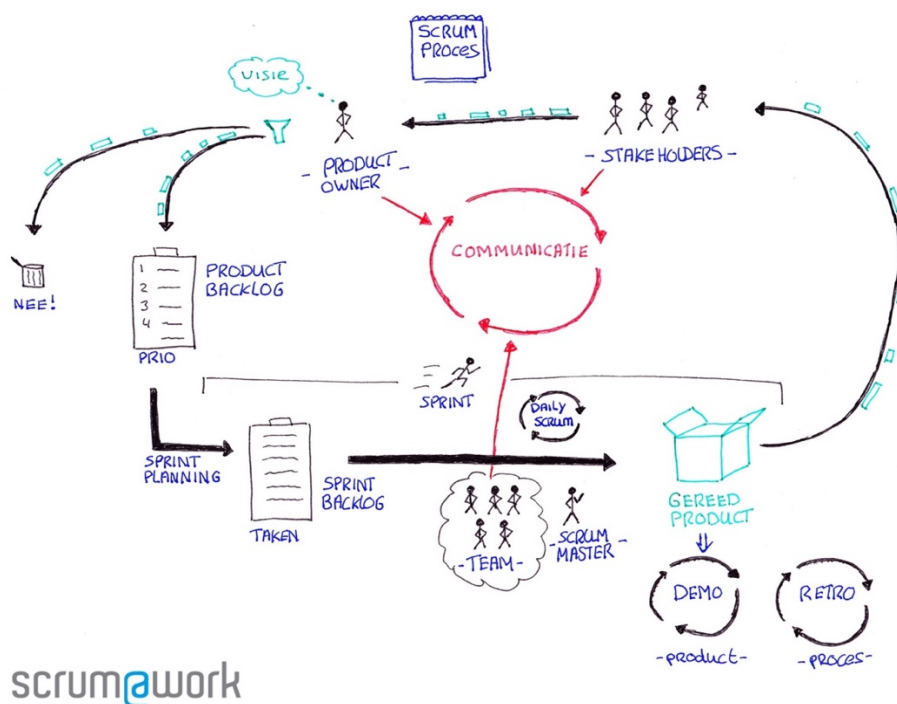
4 bijeenkomsten

1. Planningsessie
2. Daily Scrum
3. Demo
4. Retrospective

3 hulpmiddelen

1. Product Backlog
2. Sprint Backlog
3. Acceptatie criteria

Dit hoofdstuk geeft een beschrijving van het scrumproces aan de hand van onderstaand schema. Indien een woord in de tekst blauw geschreven is, komt dit terug in het schema.



figuur 3: het scrumproces

De centrale plaats binnen Scrum wordt ingevuld door de rol van **Product Owner**. Hij vertegenwoordigt de opdrachtgever of gedelegeerd opdrachtgever en heeft een duidelijke **visie** op het project. Hij kan duidelijk het belang van het project aangeven en vertellen waarom het project zo belangrijk voor de organisatie is. Vanuit diverse **stakeholders**, dat zijn de belanghebbenden ofwel personen die allemaal iets van het project willen en verlangen, krijgt de Product Owner allerlei wensen en vragen maar ook wettelijke eisen. Al deze wensen en eisen, die in de theorie van Scrum user stories worden genoemd (project story is ook een goede benaming), zet de Product Owner op één grote lijst; de **Product Backlog**. De Product Owner zet alle wensen op deze lijst in de juiste volgorde, waarbij het product met de meest toegevoegde waarde bovenaan en met het minst toegevoegde waarde onderaan staat. Onder toegevoegde waarde wordt verstaan de mate waarin de stakeholder op dit (deel)product zit te wachten en het bijdraagt aan het realiseren van de visie. Hoe belangrijker een product, hoe meer toegevoegde waarde het product heeft, hoe hoger het op de Product Backlog staat. De Product Owner is eigenaar van deze lijst, hij is de enige die daar nieuwe wensen op mag zetten en hij is de enige die de prioriteit mag veranderen. Door de veranderende projectomgeving en de daarmee gepaard gaande wijzigingen in wensen en eisen, ontstaat er een dynamische lijst. De inhoud en de volgorde van de Product Backlog kan continue wijzigen, gedurende de hele looptijd van het project. In tegenstelling tot traditioneel projectmanagement waarbij de projectscope vast staat, statisch is en bij start van het project vastgesteld wordt.

Zoals in voorgaande alinea is aangegeven, noemen we de (deel)producten op de Product Backlog user stories (of project stories). User stories hebben een vast opmaak en structuur:

als <belanghebbende, stakeholder>
wil ik <wens, actie>
zodat <doel, resultaat, waarde>

Door dit formaat te gebruiken, wordt u gedwongen na te denken voor wie u iets doet (voor welke stakeholder), wat hij/zij er mee wil gaan doen (wat de behoefte is) en waarom u dat doet (wat is de toegevoegde waarde). Het vormt a.h.w. een kort verhaaltje en stimuleert daarmee de mondelinge communicatie. User stories moeten zo duidelijk opgeschreven worden dat er ruimte overblijft voor creativiteit in de uitvoering (het "hoe"), maar klein genoeg zijn om te overzien en te begrijpen.

Een user story is een concreet product of deelproduct die een beschrijving van de "wat-vraag" vanuit de stakeholder aangeeft. Het zijn (deel)producten die concreet gemaakt kunnen worden en geen (visie)doelstellingen.

Aan iedere user story moeten acceptatie criteria gekoppeld worden. Acceptatie criteria zijn eisen waar een (deel)product aan moet voldoen. Deze acceptatie criteria zijn bedoeld om vooraf duidelijkheid te verschaffen en om interpretaties te voorkomen, zowel voor de Product Owner als voor het team dat het werk moet doen. Doordat deze criteria vooraf zijn beschreven, weet het team vooraf precies wat ze moeten doen om een product af te ronden. Binnen de literatuur van Scrum wordt de term Definition of Done gebruikt. Hoewel er duidelijk een verschil is tussen de Definition of Done en acceptatiecriteria, is het uiteindelijke doel en gebruik er van gelijk aan elkaar: criteria om te bepalen of een product "af" is. Wij hebben er voor gekozen om beide begrippen gelijk aan elkaar te behandelen.

Het daadwerkelijk maken of realiseren van deze producten wordt gedaan door een team, het zogenaamde **projectteam** of realisatieteam. Dit team is zelforganiserend en zelfsturend wat inhoud dat ze zelf verantwoordelijk zijn hoe ze hun werk organiseren. Ieder teamlid moet

betrokken zijn bij het plannen, inschatten, benoemen van blokkades, het verdelen of oppakken van taken, etc. Daar waar de Product Owner aangeeft “wat” er gedaan moet worden, bepaalt het team zelf “hoe” ze de user stories realiseren.

Het realisatieteam werkt in vaste tijdrhythmes, de zogenaamde **sprints**. De sprints hebben een vaste lengte van bijvoorbeeld twee weken. Door het werken in vaste sprints, ontstaat een workflow. Het voordeel van het werken in een flow is dat er over heel veel dingen niet meer nagedacht hoeft te worden, het gaat vanzelf of is vanzelfsprekend. Dit heeft weer tot gevolg dat u meer kunt doen in dezelfde tijd. Een sprint kan gezien worden als een soort mini project, een mini planningsperiode. Net als een groot project bepaalt u per sprint vooraf wat er gemaakt moet worden, wordt er een planning gemaakt, worden er producten gemaakt en bekijkt u of u nog op planning ligt.

In iedere sprint levert het team **afgeronde producten** op. Producten die het stempel “klaar” krijgen, geen *half werk* of producten die *bijna klaar* zijn.

Aan het begin van iedere sprint bepaalt het team welke producten van de Product Backlog ze in de komende sprint kunnen oppakken. Dat gaat op basis van de prioriteit van deze Product Backlog. Het team bepaalt zelf hoeveel producten ze denken af te kunnen krijgen. Er is geen projectleider of teamleider die opdraagt hoeveel werk ze moeten doen. Scrum gaat uit van het pull-mechanisme en niet vanuit een push-mechanisme. Dat heeft vooral te maken met het begrip *eigenaarschap* en *commitment*. Een team dat zelf de hoeveelheid werk bepaalt, voelt zich veel meer eigenaar van deze miniplanning en is daarmee meer gecommiteerd dan in de het geval een projectleider dit opdraagt. In dat geval heerst toch vooral het gevoel dat het de planning van de projectleider is en niet de planning van het team. Door het gevoel van eigenaarschap is het team effectiever en zal de miniplanning van twee wekelijkse sprints vaker gehaald worden. Het team moet bij het plannen van de sprintinhoud rekening houden met de productiviteit per sprint. Stel dat het team iedere sprint tussen vijf en zeven producten oplevert, dan is dat hun productiviteit. Neemt het team, bijvoorbeeld door toedoen van een Product Owner, stakeholder of Scrummaster, vijftien producten op in de sprint, dan raakt het team overbelast. Overbelasting (meer proberen te doen dan haalbaar is) werkt negatief op de productiviteit. Een metafoer om dit aan te duiden is een snelweg, die ook een bepaalde capaciteit heeft. Indien er bij een overvolle snelweg toch meer auto's die weg op gaan, ontstaat er file en rijdt er niemand door.

Als een sprint is gestart, dan is de inhoud van die sprint vastgesteld. In tegenstelling tot de Product Backlog, dat een dynamische en flexibele lijst is, is de Sprint Backlog een statische lijst waar tijdens de sprint geen nieuwe producten aan mogen worden toegevoegd of afgehaald mogen worden. Wijzigingen in de sprintinhoud hebben grote negatieve gevolgen voor de productiviteit van het team.

Het team wordt in alle facetten van Scrum ondersteund door de **Scrummaster**. Deze rol is verantwoordelijk voor het scrumproces, hij is er verantwoordelijk voor dat alle partijen volgens de spelregels van Scrum werken en faciliteert het gehele scrumproces. Het is een faciliterende en geen hiërarchische leider van het team. Scrum heeft heel nadrukkelijk de taken van de traditionele projectleider verdeeld over meerdere rollen. De traditionele projectleider is verantwoordelijk voor zowel inhoud als voor het proces. Scrum is van mening dat dit te veel is voor één persoon en heeft de taken verdeeld over twee rollen: de Product Owner is verantwoordelijk voor de inhoud van het project en de Scrummaster is verantwoordelijk voor het proces.

Alle producten die het team denkt te kunnen realiseren in de sprint, worden van de Product Backlog afgehaald, vertaald naar taken en op de **Sprint Backlog** gezet. Een andere benaming voor de Sprint Backlog is het Scrumbord. Het vertalen naar taken gebeurt in een meeting die **planningssessie** wordt genoemd. De Sprint Backlog is een lijst met drie kolommen: To Do – Doing –

Done (in het Nederlands: Te doen – bezig – klaar). In de kolom To Do komen alle taken te hangen die het team moet doen om het betreffende product af te krijgen. Het doel van het team is om gedurende de sprint zo snel mogelijk alle takenbriefjes van de kolom To Do in de Done kolom te krijgen.

Iedere keer als het realisatieteam bij elkaar komt (het liefst dagelijks) beoordeelt het team hoe de trend is van het gedane werk ten opzichte van de Sprint Backlog. Ze beoordelen de situatie waar ze precies staan en wat ze moeten doen om die dag zoveel mogelijk takenbriefjes naar Done te kunnen verhangen. Deze meeting heet de Daily Scrum (of Daily Standup). Deze meeting duurt maximaal vijftien minuten en ieder teamlid beantwoordt drie vragen:

1. Wat heb ik gedaan sinds de vorige keer dat we bij elkaar waren (aan welke takenbriefjes heb ik gewerkt)?
2. Wat ga ik vandaag doen? Aan welke taken ga ik vandaag werken?
3. Is er enig obstakel die mij of het team in de weg staat de taken in de sprint af te ronden?

Als iedereen deze vragen beantwoordt, weet hoeveel takenbriefjes er nog gedaan moeten worden in de resterende sprinttijd, kan het team bepalen waar het staat t.o.v. hun planning van de sprint. Een hulpmiddel hierbij is de sprint burndown. De sprint burndown is een planningshulpmiddel waar de ideale en de werkelijke planningslijn op staan. Door te bekijken of de werkelijke lijn boven of onder de ideale lijn zit, kunt u afleiden of het team vóór of achter op schema ligt. Het team weet nu of ze de inhoud van de sprint kunnen afronden en daarmee "de sprint gaan halen". En als ze het niet halen, weten ze ook waar dat aan ligt en kunnen dus acties bedenken om dat op te lossen. Omdat je als *team* verantwoordelijk bent voor het sprintdoel, ben je ook als *team* verantwoordelijk voor het oplossen van deze obstakels. De Daily Scrum is een typische Scrum inspect-and-adapt meeting: kijken waar we staan en acteren, reageren en bijsturen op de actuele situatie. De Daily Scrum verbetert de communicatie tussen teamleden, voorkomt andere bijeenkomsten en zorgt voor het maken van snelle beslissingen.

Na een sprint werken aan de producten op de Sprint Backlog, worden tijdens de **demo** de eindproducten die "af" en "klaar" zijn (die aan alle acceptatie criteria voldoen), door het team aan de Product Owner en stakeholders getoond. Er wordt aan de hand van de acceptatiecriteria getoetst of datgene wat door het team is gerealiseerd, ook daadwerkelijk het stempel "klaar" krijgt. Voor de Product Owner en stakeholders is de demo een belangrijk moment omdat er complete producten getoond worden waar ze verder mee kunnen. Het is een moment voor het team om te toetsen of ze met het project nog op de goede weg zijn, of ze nog de goede dingen doen en of er nieuwe wensen vanuit de Product Owner zijn.

De sprint wordt afgerond met de **retro**(spective). Staat tijdens de demo het product centraal, tijdens de retro gaat het over het proces. Hoe hebben wij het als team nu gedaan? Wat moeten we blijven doen maar wat moeten we veranderen om ervoor te zorgen dat de volgende sprint beter, leuker en effectiever is dan de vorige sprint? Uit de retro komen maximaal drie tot vier verbeterpunten naar voren. Zorg ervoor dat aan al deze actiepunten opvolging wordt gegeven omdat anders de kans bestaat dat er in het team een passieve houding ontstaat omdat er "toch nooit iets met de verbeterpunten wordt gedaan".

Als de retro is geweest, is de sprintcyclus afgelopen en begint direct de volgende sprint met de planningsessie. Tussen het einde van de ene sprint en de start van de volgende sprint zit geen tijd of werkdagen tussen.

Het lijkt misschien dat de afzonderlijk rollen door hun eigen taken en verantwoordelijkheden veel alleen moeten doen. Maar bij Scrum is niets minder waar: er is juist continue **communicatie** tussen alle partijen die bij het project betrokken zijn. De Product Owner staat in continue contact met de

stakeholders en helpt ze met het identificeren en beschrijven van user stories. De Product Owner praat regelmatig met het realisatieteam om vragen te beantwoorden, om nieuwe wensen vanuit de stakeholders te bespreken, om user stories aan te scherpen en om inschattingen te laten doen. Dit zijn geen vooraf ingeplande lange meetings maar spontane en ad hoc overleggen. Deze korte lijnen zorgen voor goede kennisoverdracht tussen alle partijen, voor hulpvragen maar voorkomen ook dat er verrassingen gedurende of na afloop van een sprint ontstaan.

In de praktijk zal de Product Owner vanuit de stakeholders regelmatig nieuwe wensen en verzoeken krijgen. Alleen producten die bijdragen aan de projectvisie, komen op de Product Backlog. Als een product niet bijdraagt aan de projectvisie, heeft het blijkbaar geen toegevoegde waarde en zou er geen kostbare tijd en energie aan besteed moeten worden. De Product Owner moet bij ieder verzoek van een stakeholder de afweging maken of het product bijdraagt aan de visie. Indien dit niet het geval is, zal de Product Owner "nee" tegen die stakeholder en diens verzoek moeten en kunnen zeggen.

Planning poker

Het inschatten van tijd is ontzettend moeilijk, sterker nog: daar zijn we als mensheid slecht in. Daarom heeft Scrum iets anders bedacht en schat het werk in op basis van relatieve inschattingen. Als u zou vragen hoe hoog een flatgebouw van twaalf verdiepingen is en hoe hoog een flat van zes verdiepingen, dan is op beide vragen moeilijk een antwoord te geven. Maar u kunt wel stellen dat een flat van twaalf verdiepingen twee keer zo hoog is als een flatgebouw van zes verdiepingen. En een gebouw van drie verdiepingen is weer de helft zo hoog als de zes verdiepingen flat. Zo werkt het principe van relatieve inschattingen.

Alle producten op de Product Backlog worden op deze manier door het team ingeschat. Uitgangspunt van scrum is dat degene die het uitvoerende werk doet, ook de inschatting doet. Zodra er door de Product Owner een nieuwe wens op de Product Backlog zet, wordt deze ingeschat door het team. Deze puntenscore is belangrijk voor de Product Owner omdat het input vormt voor het bepalen van de totale omvang van het project en voor de prioriteit.

De inschatting wordt uitgedrukt in Story Points. Om user stories in story points in te schatten, heeft Scrum een methode ontwikkeld die planning poker heet. De methode maakt gebruik van planning poker kaarten die zijn opgebouwd uit de cijfers 0 – ½ - 1 – 2 – 3 – 5 – 8 – 9 – 13 – 20 – 40 en 100. Deze cijferreeks is grotendeels gebaseerd op de fibonaccireeks. Deze reeks wordt opgebouwd door twee opeenvolgende getallen te nemen die te samen het daarop volgend getal vormen. Voorbeelden uit de reeks:

- 1 + 2 = 3,
- 3 + 2 = 5,
- 5 + 3 = 8,
- etc

Tot dat u bij 13 + 8 = 21 uitkomt. Het is geen toeval dat planning poker 20 aangeeft en geen 21. Het zou namelijk bij een complex product betekenen dat u precies kunt aangeven dat het 21 story points is. Maar naarmate de getallen in de reeks groter worden, neemt de onnauwkeurigheid ook toe. Om deze onnauwkeurigheid te benadrukken, bestaat het getal 21 niet maar gaat de reeks verder met 20, 40 en 100. Onnauwkeurigheid betekent in dit geval dat we niet precies kunnen aanduiden hoe groot iets is.



figuur 4: planning poker kaarten

Volgens welke regels verloopt het inschatten met planningpoker:

- 1. Product Owner leest user story voor en geeft toelichting*
- 2. Vragen stellen/discussie over user story*
- 3. Ieder teamlid trekt een kaart en legt hem omgekeerd neer*
- 4. Tegelijk alle kaarten omdraaien*
- 5. Alle schattingen gelijk -> volgende user story*
- 6. Schattingen ongelijk -> hoogste en laagste geven toelichting, daarna opnieuw naar 3 (terug naar 2 indien de user story nog onduidelijkheden omvat)*

Het gebruik van het planning poker principe heeft grote voordelen. De inschattingen a.d.h.v. planning poker zijn nauwkeuriger dan absolute inschattingen in uren. Dat komt omdat de fibonacci reeks in de natuur en natuurkunde op heel veel plaatsen voorkomt en daarmee dicht bij ons begripkader ligt. Kijk maar eens naar uw vinger: de lengte van het topje en het middelste kootje vormen samen de lengte van het onderste kootje. Daardoor begrijpen we instinctief de reeks beter en begrijpen we wel duidelijk het verschil tussen vijf en acht maar niet tussen, bijvoorbeeld, zeven en acht.

Maar wat is nu een story point precies? We drukken de producten op de Product Backlog uit in complexiteit en inspanning. M.a.w. hoe complex ofwel hoe moeilijk is het en hoeveel inspanning kost het om een specifiek product te realiseren. De combinatie van complexiteit en inspanning wordt in story points uitgedrukt. Als het product A drie story points is en product B is acht story points, dan betekent dat product B ongeveer drie keer zo complex is om te maken. Alle producten zet u op deze manier af op één centraal gekozen product: de referentie user story. Alle inschattingen die u doet, vergelijkt u met deze user story. U stelt uzelf allereerst de vraag of het in te schatten product complexer of minder complex is dan het referentie product? Om u vervolgens af te vragen hoeveel complexer of minder complex het dan is? Zo komt u tot een eigen inschatting.

Naast dat het gebruik van planning poker nauwkeuriger is, gaat het inschatten ook sneller dan in het geval er een inschatting gemaakt moet worden op basis van uren. Een niet te onderschatten voordeel is het feit dat er gedurende het inschatten door het team al inhoudelijk over het product wordt gesproken wat ongemerkt zorgt voor kennisoverdracht over het product.

Eerder hebben we het begrip eigenaarschap benoemd. Dit principe van eigenaarschap is ook voor de inschattingen van toepassingen doordat bij Scrum de inschattingen gedaan worden door degenen die het werk moeten doen. Door het realisatieteam en niet door een projectleider. Hierdoor wordt het de "inschatting van het team" en niet die van een projectleider. De producten worden als team ingeschat en niet door één persoon. Er volgt dus als eindresultaat één teaminschatting en niet individuele, persoonlijke inschattingen.

De inschattingen in story points kan door de Product Owner gebruikt worden in het bepalen van de prioriteit. Sommige producten zijn klein en sommige zijn groot. Andere producten zijn heel belangrijk en andere producten zijn puur wensen. Er is geen relatie tussen beide aspecten. Ter verduidelijking twee voorbeelden.

- Als er twee producten A en B zijn die evenveel toegevoegde waarde bieden, maar product A is tweemaal zo complex als product B, maak dan eerst product B. De PO zal product B daarmee een hogere prioriteit geven en hoger op de Product Backlog zetten.
- Als producten C en D even complex zijn om te realiseren (gelijk aantal story points) maar product C biedt meer toegevoegde waarde dan product D, doe dan eerst product C. De Product Owner zal product C hoger op de Product Backlog zetten dan product D.

Om de juiste prioriteit te kunnen bepalen moet van de producten de inschatting en de toegevoegde waarde bekend zijn.

Stakeholders

Een project is nooit een doel op zich, maar een middel om tot een doel te komen. Iedereen die als belanghebbende bij het doel van een project betrokken is, bestempelen we als stakeholder. Een stakeholder is een persoon die invloed op of belang heeft bij het project. Stakeholders en stakeholdermanagement zijn taken primair voor de Product Owner. Het helpt de Product Owner om structuur in deze groep van belanghebbenden aan te brengen.

Daartoe is onderstaand diagram en aanpak een handig hulpmiddel.

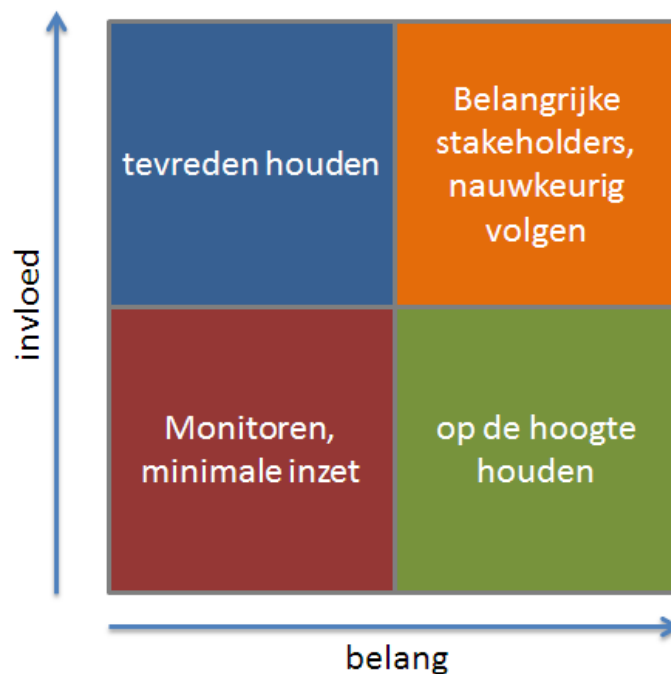
Stap 1: identificeren van de stakeholders

Stap 2: classificeren van de stakeholders

Bij stap 1 identificeert u zo alle stakeholders die bij het project betrokken zijn. In stap 2 classificeert u iedere stakeholder uit stap 1 op basis van de criteria invloed en belang. Voor deze stap kunt u uzelf de volgende vragen stellen:

- Invloed: heeft de stakeholder de mogelijkheid om met zijn houding en gedrag een project te vertragen of te versnellen?
- Belang: wie heeft er het meeste belang bij het slagen van het project of anders geformuleerd, wie gaat in de problemen komen als een project mislukt?

Plaats vervolgens de uitkomsten in het diagram in figuur 5.



figuur 5: stakeholdersmatrix

De belanghebbenden in het kwadrant rechtsboven vormen de belangrijkste stakeholders. Zij geven de input aan de product owner over hun eisen en wensen aan het project. Zij zijn de groep die het meeste belang hebben bij het welslagen van het project maar hebben tegelijkertijd de meeste invloed op het project. Deze groep stakeholders dient de product owner heel nauwkeurig te volgen, goed contact mee onderhouden om (veranderende) wensen vroegtijdig te onderkennen.

De groep stakeholders die een grote invloed heeft maar een kleiner belang, moet door de Product Owner tevreden houden zodat hun invloed positief voor het project ingezet wordt. De

stakeholders met een hoog belang maar minder invloed moet de Product Owner o.a. op de hoogte worden gehouden over de ontwikkelingen en de voortgang van het project. De groep stakeholders met een laag belang en weinig invloed moet door de Product Owner gemonitord worden om gevoel te houden met hun houding en gedrag t.o.v. het project, maar de inzet en tijdsbesteding is minimaal.

Bij de start van een scrumproject levert de groep stakeholders in het kwadrant rechtsboven de wensen en eisen die voor het project het meest belangrijk zijn. Zij leveren de input voor de Product Owner om de eerste versie van de Product Backlog samen te stellen. Bij het begin van de project gaat de Product Owner eerst bij deze groep stakeholders informeren naar de wensen en eisen.

Scrum in de praktijk met Jip en Janneke

Om de werking van Scrum in Jip en Janneke taal uit te leggen, willen we dit graag vertellen middels een verhaal van Jip en Janneke: Jip en Janneke en de speeltuin

Jip en Janneke en de speeltuin



figuur 6: Jip en Janneke maken een speeltuin

Op een dag komt Jip bij Janneke. Jip wil heel graag een hele mooie speeltuin. Met van alles erin. Janneke kan dit wel maken maar ze wil eerst weten wat er dan allemaal in die speeltuin moet komen. 'Een zwembad, en een glijbaan, en een zandbak en nog veel meer' zegt Jip. Janneke schrijft het allemaal op in een mooi plan. Jip leest het even door en zegt tegen Janneke dat ze het mag gaan maken maar dat hij eerst alles tot in de puntjes wil zien. Janneke gaat aan de slag met heel veel stoepkrijt en laat aan het eind van de week alles zien.

Jip vind het allemaal heel mooi. Maar hij wil nog meer weten. "Hoe hoog wordt de glijbaan. Hoe warm het zwembad. En wat voor kleur krijgt het zand in de zandbak" vraagt Jip. Janneke weet het niet dus verzint ze van alles. "twintig meter hoog. En heel erg warm. En het zand is rood". Jip gaat ermee akkoord en geeft Janneke al zijn spaarcentjes. "Over twee weken kom ik kijken" zegt hij en gaat weg. Janneke gaat aan de slag. Ze heeft wat vriendjes uit de buurt gevraagd om te komen helpen. "Maar je kunt helemaal geen glijbaan zo hoog bouwen voor die centjes, en hoe warm is warm. En rood zand is heel erg vies voor je kleren" zeggen ze. "Dat kunnen we nooit zo maken" Janneke is radeloos. Ze had het Jip zo beloofd en als ze terug gaat wordt hij misschien wel boos. Dus besluit ze aan de slag te gaan zo goed als het gaat. Ze beloofd haar vriendjes heel veel ijsjes. De extra centjes voor de glijbaan haalt ze uit haar eigen spaarpot. Het water wordt gewoon lekker warm, en bij de rode zandbak geeft ze wel extra wasmiddel mee voor Jip zijn moeder. Als Jip na twee weken terugkomt weet hij niet wat hij ziet. Alles is er maar niet zoals hij in gedachte had. De twintig meter hoge glijbaan durft hij echt niet op. Het zwembad is te warm. En wat zal zijn moeder boos worden als ze de rode vlekken op zijn broek ziet. Jip is boos. En hij wil dat Janneke de speeltuin opnieuw gaat

maken. De vriendjes wilde niet meer helpen want het is toch niet hun eigen speeltuin en ze moeten op tijd thuis zijn. Janneke moet huilen en rent naar haar moeder.

Jammer dat Jip en Janneke niet echt samen met elkaar gingen spelen. Jip had samen met Janneke kunnen bedenken hoe het moest worden. Maar ook gelijk met alle vriendjes erbij. Want Timmy zou een prachtig idee hebben gehad voor de glijbaan. Een waterglijbaan. En jip had gelijk het water kunnen voelen en kunnen zeggen dat het kouder mocht. En de moeder van Jip had het zand kunnen afkeuren en met mooi wit zand kunnen komen. Dat zou ook mooier staan bij de waterglijbaan en alles tot een heel mooi klein paradijsje kunnen maken. Tijdens het maken van de speeltuin zouden ze ook achter zijn gekomen dat Jip eigenlijk niet kan zwemmen. Dus hadden ze het water kunnen vervangen voor ballen. Die ook nog eens minder duur zouden zijn zodat ze centjes zouden overhouden voor een wip. Jip en Janneke hadden samen met hun vriendjes de mooiste speeltuin kunnen maken van de buurt. Met zelfs een echte schommel. Want de boom die ze tijdens het maken tegenkwamen had een prachtige tak, en Janneke had nog een stuk touw en een oude binnenband.

Scrum zorgt ervoor dat je samen gaat spelen om een zo mooi mogelijke speeltuin te maken. Door goed samen te werken en goed met elkaar te praten kom je tot oplossingen. Er komen nieuwe ideeën naar boven waar nog niemand eerder aan gedacht had. Een schommel en een ballenbad. Jip kreeg de speeltuin die hij eigenlijk wilde hebben maar dit van te voren niet had kunnen bedenken. En Janneke kon haar spaarvarken heel laten. En de vriendjes uit de buurt hadden de grootste lol in het maken van de tuin. Ze leerde heel veel van elkaar en ook loste ze samen alles op. Janneke had haar vriendjes de kans gegeven om zelf met oplossingen te komen zolang haar spaarvarken maar niet kapot hoefde. En de vriendjes hadden Jip weten te overtuigen en bedachten nog mooiere oplossingen. Het maken van de speeltuin was echt samenspel geworden en iedereen was blij.

Tot slot

Wij wensen u tot slot na het lezen van deze scrumgids heel veel scrumplezier. De theorie van Scrum is vrij eenvoudig maar het goed toepassen vergt meer tijd en inspanning. Vergelijk Scrum met schaken of dammen: daar zijn ook niet veel spelregels maar er zijn onuitputtelijk veel mogelijkheden om het spel te spelen en een wedstrijd winnen, kost veel inspanning en oefening. Mocht u na het lezen van deze scrumgids nog vragen hebben, neem dan contact met ons op via info@scrumatwork.nl of via 085 – 864 02 40.

Bijlage 1: begrippenlijst

4 rollen

Product Owner

- Visie.
- Bepaalt wat er op de Product Backlog komt en in de juiste volgorde (prioriteit) om de doelen van het project op de beste manier te behalen. Is eigenaar van de Product Backlog.
- Inventariseert de wensen en eisen van de stakeholders (stakeholders management).
- Geeft aan "wat" het team moet doen.
- Kaderstellend en richtinggevend.
- Waarde toevoegen.

Scrummaster

- Begeleider van het scrumproces.
- Dienend leider voor het team. Is geen hiërarchische rol/functie.
- Verantwoordelijk dat Scrum door iedereen wordt begrepen en goed wordt uitgevoerd.
- Coachen van het team om zelforganiserend en zelfsturend te werken (teammanagement).
- Blokkade opruimer.
- Organiserend.
- Doel is om het team continue te laten verbeteren.

ProjectTeam

- Maakt de producten, verzorgt het echte werk.
- Bepaalt het "hoe".
- Zelforganiserend en zelfsturend.
- Geen hiërarchie binnen het team.
- Is als geheel verantwoordelijk voor halen van het resultaat.
- Eigenaarschap.
- Continue verbeteren.

Stakeholder

- Iedere betrokkene bij het project:
 - die een belang heeft bij het project
 - of invloed heeft op het project
- Levert input (eisen en wensen) aan de Product Owner en daarmee aan de Product Backlog.
- Belangrijke gesprekspartner voor Product Owner.

4 meetings

Planningsessie

- Eerste meeting in een sprint.
- Team bepaalt de inhoud van de sprint.
- O.b.v. de prioriteit van de Product Backlog.
- Maken van de Sprint Backlog met de taken.

Daily Scrum

- Meeting iedere keer als team bij elkaar komt.

- Bespreken voortgang van de sprint en belemmeringen die de voortgang in gevaar brengen.
- Ieder teamlid beantwoordt deze 3 vragen:
 - 1) wat heb ik gedaan sinds de vorige Daily Scrum?
 - 2) wat ga ik vandaag doen?
 - 3) Zie ik enig obstakel om het sprintdoel te behalen en welke hulp heb ik daarbij nodig?
- Staand bij Sprint Backlog, maximaal 15 minuten.
- De Daily Scrum verbetert de communicatie, voorkomt andere bijeenkomsten en zorgt voor het maken van snelle beslissingen.

Demo

- Team bespreekt wat ze hebben gedaan.
- Team toont de resultaten van de sprint aan de Product Owner en eventueel de stakeholders.
- Gaat over het product.
- Product Owner kan beoordelen of team nog met de juiste dingen bezig is.
- Na afloop van iedere sprint is er een demo.

Retrospective

- Team bespreekt hoe ze het hebben gedaan.
- Gaat over hoe het proces is verlopen.
- Team bespreekt wat ze in het proces kunnen of moeten aanpassen om de volgende sprint beter te laten verlopen.
- Vasthouden wat goed gaat.
- Na afloop van iedere sprint is er een retrospective.

3 hulpmiddelen

Product Backlog

- De to-do lijst van het project: welke activiteiten moeten er allemaal uitgevoerd worden en in welke volgorde?
- Beschrijft alleen het "wat".
- Geprioriteerd o.b.v. toegevoegde waarde (en inschatting in story points).
- Is een dynamische lijst (er kunnen onderdelen bij komen, afvallen en de volgorde kan veranderen). Het opstellen van de backlog gebeurt in dezelfde regelmaat als de sprints. Hierdoor is de backlog altijd actueel.

Sprint Backlog

- Andere benaming: het scrumbord (To Do – Doing – Done).
- To-do lijst van waar tijdens huidige sprint aan gewerkt wordt.
- Bevat de taken om een Product Backlog item af te maken.
- Beschrijft het "hoe".
- Geeft de inhoud en de voortgang van de sprint weer.

Acceptatie criteria

- Acceptatie criteria zijn regels waar een product aan moet voldoen om "klaar" te zijn.
- Mogen geen ruimte voor interpretatie geven. Geen aannames.
- Acceptatie criteria worden vooraf bepaald zodat het team vooraf begrijpt wat ze moeten doen.

Diversen

User Story

- Deelproduct van het project.
- In verhalende vorm vanuit stakeholder beschreven wens of eis.
- Andere benaming is project story.
- Formaat:
als <belanghebbende, stakeholder>
wil ik <wens, actie>
zodat <doel, resultaat, waarde>
- Bevordert de mondelinge communicatie.

Story point

- Geeft de relatieve inschatting van de complexiteit en inspanning in punten weer.
- Voor de punten toekenning wordt planning poker met de fibonacci reeks gebruikt.

Planning poker

- Proces om te komen tot een relatieve inschatting van de complexiteit en inspanning.
- Iedere user story wordt ingeschat door het team.

Sprint

- Een vaste timebox waarin het team in een vast ritme user stories omzet naar een gereed producten.
- Een sprint kan gezien worden als een miniproject. In dit mini projecten wordt een planning gemaakt, wordt er gemaakt en worden er afgeronde producten opgeleverd.
- Sprints bestaan uit een Sprint Planningsbijeenkomst, Daily Scrums, het uitvoerende werk, de demo en de retrospective.
- Een nieuwe sprint start direct nadat de vorige sprint is afgesloten.
- De inhoud van een sprint mag niet tijdens de sprint gewijzigd worden.
- Een sprintlengte van 2 weken is in de praktijk een goede sprintlengte gebleken. Maar kies een vaste sprintlengte die niet te lang is.

Sprintdoel

- Het sprintdoel van iedere sprint is om alle takenbriefjes op het scrumbord voor het einde van de sprint op Done te krijgen zodat er alleen afgeronde producten worden opgeleverd.

Bijlage 2: Scrum binnen en buiten IT

Scrum wordt op grote schaal binnen de softwareontwikkeling toegepast. Ken Schwaber en Jeff Sutherland hebben Scrum ontwikkeld om juist binnen het IT domein toe te passen. Omdat Scrum ook van toegevoegde waarde is buiten het IT domein en de software ontwikkeling, is deze scrumgids voor deze doelgroep herschreven.

Er zijn echter grote verschillen in het toepassen van Scrum binnen of buiten de softwareontwikkeling. Niet alleen de invulling van de methode is afwijkend, ook de manier hoe project- en teamleden reageren op en handelen naar de scrumtheorie. Om Scrum succesvol toe te passen, is het verstandig je bewust van deze verschillen te zijn. Onderstaand schema geeft een aantal verschillen weer:

	IT domein	Buiten IT domein
Niveau user stories	veel details	Hoog over, minder details
Aantal user stories	Heel veel	Relatief weinig (<10 per person, soms slechts 3 tot 4 per person)
Hersenhelft	Linker kant, beta's (logica, detail, realiteit, praktisch, rekenen en wetenschap, begrijpen, weten)	Rechter kant, alpha's (het grote geheel, impulsief, verbeelding, symbolen en plaatjes, betekenis, overtuiging, waardering, ruimtelijk inzicht, fantasie)
Focus	Focus op functionaliteit	Focus op resultaat (outcomes)
Gezamenlijk doel	Gezamenlijk (project)doel	Gedeeld doel minder duidelijk (wel op organisatieniveau maar niet in teamverband)
Voorbeelden		Voorbeelden uit ICT helpen niet om ideeën te verduidelijken aan professionals buiten de softwareontwikkeling.
Structuur	Werk is gestructureerd en kan vaak zelfstandig uitgevoerd worden.	Werk is minder gestructureerd en doorlooptijden worden beperkt door afhankelijkheden van andere groepen.
Zelfsturend	Werknemers zijn vaak zelfsturend en initiatiefrijk	Teamleden zijn meer gewend opdrachten te krijgen.
Transparantie	Teamleden zijn meer gewend om verantwoording af te leggen. Transparantie is meer gewoonged.	Teamleden voelen zich wellicht minder gemakkelijk als er meer transparantie mbt hun werk is.
Kennisniveau teamleden	Teamleden liggen wat kennisniveau dicht bij elkaar (allemaal HBO en allemaal software ontwikkelaars en consultants)	Team heeft meer variëteiten in ervaringen en kennisniveaus.
Individueel doel	Individueel doel is vaak ondergeschikt aan projectdoel. Consultants worden ingehuurd voor uitvoeren project en dus voor projectdoel.	Individen hebben wellicht minder belang bij een teamdoel als ze een focus op hun eigen belang en doel hebben.