DE SAMENWERKENDE ANALIST

DE ROL VAN DE ANALIST IN AGILE TEAMS

Praktische tips voor een optimale samenwerking

DiVetro
Digitale Vooruitgang

COLOFON

Dit e-book is een publicatie van **DiVetro**, een onafhankelijk strategisch adviesbureau voor IT-gerelateerde vraagstukken. De wereld positiever maken door te werken aan de digitale vooruitgang bij onze opdrachtgevers, dát is onze ambitie. Wij zijn vooral actief op projecten bij de (semi-)overheid, in de gezondheidszorg en in het onderwijs.

Auteurs
Dennis Geluk
Marco Balvers
Marco van Dop
Ronald Koenis
Selmar van Egmond
Stanley Lachman
Tijn van Wegen

Uitgever
DiVetro

INHOUDSOPGAVE

DiVetro
Digitale Vooruitgang

INLEIDING

Als analist werk je in agile teams samen met collega's die verschillende taken en verantwoordelijkheden hebben. Er is een gemeenschappelijk doel met een aparte focus voor elke rol. Een goede onderlinge samenwerking en communicatie is cruciaal om op een efficiënte manier tot een passende (software)oplossing te komen.

Wij, analisten bij DiVetro, hebben onze inzichten en praktische tips over samenwerking in dit e-book gebundeld. We gebruiken deze inzichten elke dag en we hopen dat ze je kunnen helpen bij de uitvoering van jouw analyse-klus.

We delen graag onze tips voor een goede samenwerking tussen analisten en:

- Product Owners met hun focus op het waarom en waarom nu;
- Architecten: de kader-gevers met een focus op de lange termijn;
- UX designers: zij bepalen het uiterlijk van de oplossing;
- Ontwikkelaars, die zich richten op hoe de oplossing werkt;
- Testers valideren dat de oplossing ook echt de business helpt;
- Beheerders, die waarborgen dat de oplossing past in de informatievoorziening;
- En ook andere analisten, die samen bepalen wat de oplossing moet gaan doen.

Veel leesplezier!

Rollen in agile teams

Agile of Scrum teams kennen alleen de rollen: Product Owner, Scrum Master en Developer (**Scrum Guide**) en geen rollen als: analist of tester.

In het ideale geval is iedere Developer ***T- of π-shaped***. en is dus van meerdere markten thuis. In onze dagelijkse praktijk heeft iedere Developer meestal een eigen specialiteit.

Als we het in dit boek hebben over een ***analist***, een ***tester*** of een ***beheerder***, dan bedoelen we een Developer met dit vakgebied als eigen specialiteit.

DiVetro
Digitale Vooruitgang

1. De samenwerking tussen analist en Product Owner

De analist verzamelt wat de wensen van de business zijn, structureert deze en legt ze begrijpelijk vast. De Product Owner bepaalt waarom het team nu zo'n business-wens moet realiseren. Deze samenwerking zorgt voor een (software)oplossing die de meeste waarde oplevert voor de business. In dit hoofdstuk geven we praktische tips voor, tijdens en aan het eind van een sprint uit Scrum.

De Product Owner en de analist

In Scrum heeft de Product Owner een zware rol in het samenstellen van de product backlog. Hij bepaalt de stories en in welke volgorde het team ze zal realiseren. Dat vereist een brede kennis van business en IT en de kunde om een story helder te formuleren. Er zijn natuurlijk Product Owners die dat allemaal in zich hebben, maar in de praktijk zien we dat de analist hem vaak helpt bij deze taken.

De analist is binnen Scrum als lid van het ontwikkelteam verantwoordelijk voor het ontwerp en de documentatie van het systeem. Als Scrum wordt opgeschaald – ongeacht of dat nu SAFe, Nexus of LeSS is – dan zien we dat analisten ook worden ingezet om het grotere geheel te bepalen en te bewaken.

Het analistenteam: een anti-patroon!

Analistenteams die los van de ontwikkelteams werken, zijn een **Scrum anti-patroon**. Het geeft een groot risico op te veel en te vroeg uitwerken en ook kennisverlies bij de overdracht. In veel gevallen zal het werken als een waterval en dat is een veel minder flexibel voortbrengingsproces!

Praktische tips bij het voorbereiden van een sprint met de Product Owner

Wil je als analist de sprint optimaal (efficiënt en zonder *waste*) voorbereiden? Dan kunnen deze tips je zeker helpen:

- Voorzie in en bewaak het "grote plaatje". Een Product Owner krijgt van alle kanten informatie en wil die in een groter geheel kunnen plaatsen. De analist is de uitgelezen persoon om hem daarbij te helpen. Hij kan de Product Owner immers helpen om te bepalen welk deel nu het meeste waarde oplevert. DiVetro heeft daarvoor trouwens handige hulpmiddelen.

- Als analist ben je de vertaler van de business-taal van de Product Owner en de IT-taal van het team. Neem die rol serieus en ontwikkel begrip om spraakverwarring tussen beide werelden te voorkomen. Maak je bekend met het jargon van de techniek en hanteer een begrippenlijst of glossary voor de termen uit de business.

- Als je als analist naast stories bijvoorbeeld ook features, epics of thema's moet uitwerken voor de Product Owner, zorg dan altijd voor samenhang en behoud van het grotere geheel. Losse stories kunnen ondersneeuwen op de backlog of leiden tot een grote lijst aan stories waar de Product Owner niet meer doorheen komt. Breng altijd een tracering aan naar de epic/feature en splits alleen *als* het echt nodig is *op het moment dat* het echt nodig is.

- Als een Product Owner overbelast is dan is het verleidelijk voor de analist om op te treden als Product Owner by proxy. In dat geval doet de Product Owner het werk richting stakeholders. De analist richt zich dan op het team en bepaalt bijvoorbeeld de prioriteit. Dit ontlast de Product Owner mogelijk wel, maar het geeft een groot risico dat de analist niet de beste volgorde kiest.
  Het is beter om de Product Owner in zijn rol te laten en hem op basis van werkafspraken bij te staan in het uitwerken van stories. Dat kan bijvoorbeeld door het verfijnen van de acceptatie-criteria, het toetsen van externe vereisten of het leveren van input voor planningstools, zoals WSJF.

- Houd rekening met de behoefte van zowel de Product Owner als het ontwikkelteam. Waar de Product Owner vaak naar de grote lijnen kijkt heeft het team meer behoefte aan details. Dit kan je oplossen door de details los of in bijlagen te bewaren, zodat de story zelf kort kan blijven. Goede templates en richtlijnen helpen hierbij het resultaat consistent te houden.

De samenwerking tijdens de sprint

Als een story in de sprint komt, dan is het de verantwoordelijkheid van de analist om de story (net) helder genoeg te krijgen om de story te laten bouwen en testen. In de uitwerking komen soms nieuwe keuzes naar boven, die de analist met de Product Owner afstemt.

In Scrum is er ruimte om ook tijdens de sprint met de Product Owner over de scope te onderhandelen. Hiermee kunnen de analist en Product Owner voorkomen dat situaties die niet voorkomen in de praktijk, in scope blijven en tot *waste* leiden.

Onze praktische tips voor de analist

- Voorkom dat de Product Owner analyse-klussen op de product backlog plaatst. Analyse-stories leveren geen business value op, maar wel *waste*, aangezien je vast meer uit-analyseert dan het team die sprint nodig heeft. Streef ernaar om het analyse-werk als taak van de realisatie van een story uit te voeren. Zo loopt het ontwerp niet voor. Bovendien kan je zo veel makkelijker inspelen op voort-schrijdend inzicht tijdens de sprint.

- Laat je ontwerp reviewen door de Product Owner, of beter nog: neem ontwerpbeslissingen met hem door. Hoe eerder ontwerpfouten of misverstanden aan het licht komen, hoe beter. Het mag niet zo zijn dat de Product Owner je ontwerp niet begrijpt. In dat geval moet je iets doen aan de kennis van de Product Owner of aan de manier waarop je je ontwerp aan hem presenteert (of aan je eigen kunde).

Tips aan het einde van de sprint

In de sprint review zal de analist de business context van een oplevering geven. Een Product Owner kan die aanvullen vanuit zijn achtergrond.

Praktische tips voor de analist

- Zorg dat stories vóór de review zijn goedgekeurd door de Product Owner. Spreek zowel met hem als met de tester af welke tests moeten zijn gedaan en controleer de uitvoering. Daarmee voorkom je ongewenst gedrag van de oplevering tijdens de sprint review.

  Bij de sprint review kan de Product Owner zich concentreren op aanvullende wensen of op nieuwe inzichten die stakeholders aanleveren. Dat is zijn rol. De analist moet duidelijk maken hoe de oplevering werkt, zodat de stakeholders kunnen aangeven of dat volgens hen goed genoeg is.

- Als teamlid zal je meediscussiëren over toekomstige stories. Als er nieuwe wensen bij een sprint review naar boven komen, dan is het de taak van de analist om die in het grotere geheel te plaatsen. Zo krijgen de Product Owner en stakeholders een beter inzicht in de plek en prioriteit van nieuwe stories.

Samenwerking met de Product Owner

Conclusie

Binnen een Scrum team is de analist vaak een eerste aanspreekpunt voor de Product Owner.

De Product Owner gaat over het **waarom** en geeft de context, zodat de analist kan bepalen **wat** daarvoor nodig is.

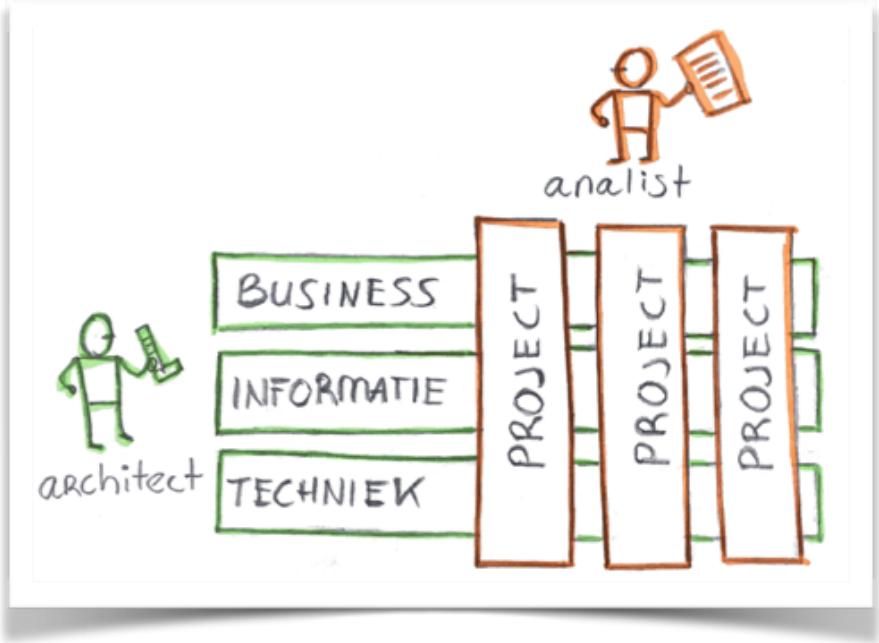
Dankzij deze samenwerking kan het team de software ontwikkelen die de meeste waarde voor de business oplevert.

2. De samenwerking tussen analist en architect

Wellicht heeft de (IT-)architect van alle rollen nog de meeste overeenkomsten met de analist. Beiden kijken immers naar informatie, bedrijfsprocessen, realisatieaspecten en vaak ook specifiek naar IT. In de praktijk zien we nog vaak dat de architect op afstand staat van het project. Ons pleidooi is om de samenwerking met de architect actief op te zoeken en te onderhouden.

De architect en de analist

Het doel van architectuur is om de informatievoorziening van de organisatie zó vorm te geven dat deze op lange termijn aansluit op de bedrijfsdoelstellingen. Veranderingen worden vanuit een bredere scope, vaak over meerdere projecten heen, door de architect gestuurd op basis van doordachte visie en kaders.

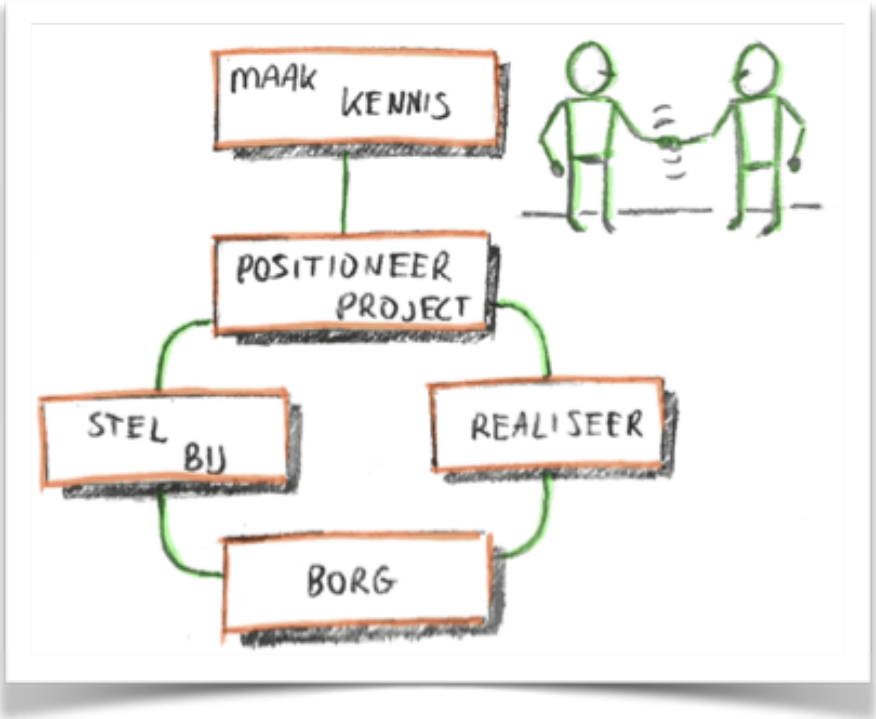


De analist richt zich meer op het project- of teamverband en kijkt naar de beste invulling van een specifieke doelstelling voor het leveren van optimale bedrijfswaarde.

De architect wordt vaak vergeleken met de bouwarchitect die alle *ins* en *outs* weet van constructie, omgeving en materialen. In deze vergelijking is de analist dan de binnenhuisarchitect. Daar waar de architect de muren van het huis neerzet, bepaalt de analist de beste toepassingen en indeling van de ruimte, met wellicht nog een kleine verbouwing tot gevolg.

Praktische tips in de samenwerking met de architect

We hebben onze tips ingedeeld aan de hand van de momenten waarop de analist en de architect elkaar tegenkomen.



1. Maak kennis met de organisatie

Als analist heb je bij de meeste veranderingen te maken met bestaande processen, systemen en toekomstplannen. Daarom is het belangrijk dat je je als analist verdiept in de organisatie. De referentiearchitectuur (hoe is het nu én wat is het toekomstbeeld?) helpt hierbij. Denk bijvoorbeeld aan processchema's op diverse niveaus, een informatiemodel en de samenhang van ondersteunde informatiesystemen.

Om wegwijs te worden, is het belangrijk kennis te maken met de architect(en) in het domein waar jij werkzaam bent. De architect heeft veel kennis van de organisatie en kan je inzicht geven in de conventies, terminologie en tradities van het bedrijf.

Als analist heb je zelf meestal ook kennis van architectuur. Een vaak voorkomende valkuil is dat jij als analist je kennis promoveert tot standaard en jezelf niet verdiept en niet of te weinig luistert. Je wilt echter het wiel niet opnieuw uitvinden, maar juist aansluiten bij je omgeving. Neem dus vooral de tijd om je eigen te maken wat er al is. Verbeteren kan daarna altijd nog.

2. Positioneer het project binnen het grotere geheel

Bij aanvang van het project stel je kaders en bepaal je de benodigde architectuur voor je projectdoelstelling. Hierbij toets je of de beoogde verandering past in de doelarchitectuur.

Wanneer architectuur op onderdelen nog ontbreekt, is verhoogde betrokkenheid van de architect nodig. Plan in zo'n geval een actie om de architectuur uit te werken, zodat er een gedeelde visie en gezamenlijk uitgangspunt ontstaat. Hiermee voorkom je onduidelijkheid en vertraging tijdens de realisatie van het project of de ontwikkeling van een 'wegwerpproduct'.

Architectuur is vaak ambitieus en een meerjarentraject. De organisatie geeft hier in fasen invulling aan, waarbij jouw project een bijdrage levert. Bepaal daarom duidelijk welke bijdrage je project in dit grotere geheel levert en waar de projectgrenzen liggen.

Architectuur is niet in beton gegoten, zeker niet in innovatieve organisaties. Maak daarom vooraf goede afspraken over de rolverdeling en het proces van bijstelling. Hiermee voorkom je vertraging in de besluitvorming en misvattingen tijdens de realisatie.

3. Realiseer bedrijfswaarde

Het realiseren van bedrijfswaarde met behulp van architectuur levert langetermijn- en herbruikbare oplossingen op. Het mes snijdt aan twee kanten: de business is geholpen en heeft hier nog lang profijt van.

Het lukt niet altijd om direct vanaf de start bedrijfswaarde te leveren. Bijvoorbeeld omdat de architectuur in het begin van een project vrij grote impact kan hebben op de benodigde inspanning om een oplossing binnen die architectuurkaders te realiseren. In zo'n geval investeer je eerst, voordat je de vruchten plukt. Zie dit niet als een 'IT-feestje', maar communiceer dit samen met de architect expliciet naar je business, zodat keuzes transparant zijn.

En vooral: werk agile! Het venijn zit vaak in de details en het kan altijd gebeuren dat een oplossing toch niet past in de architectuur. Hier kom je het liefst zo snel mogelijk achter. Of iets op papier past, is altijd nog iets anders dan de praktijk. Daarom raden we ook aan om voortdurend te integreren in het landschap, waarmee je bewijst dat de bedachte oplossing in samenhang werkt.

4. Borg voor meer snelheid, wendbaarheid en effectiviteit

Met een stevige borging zorg je ervoor dat de modellen, en hiermee waardevolle kennis, behouden blijft voor de organisatie. Tegenwoordig is documenteren in veel organisaties het kind van de rekening. Wij denken dat deze borging van kennis juist cruciaal is voor het maken van snelheid, wendbaarheid en effectiviteit van organisaties.

Architectuurmodellen waar je als analist zeker een bijdrage aan kunt leveren zijn het informatiemodel, het bedrijfsprocesmodel en de applicatiearchitectuur. Vaak zijn deze er al op hoofdlijnen en kun je detailleren, bijvoorbeeld door attributen toe te voegen aan een informatiemodel of procestappen te beschrijven binnen een werkproces. Maak hierover concrete afspraken met de architect. En vergeet niet hem de toegevoegde items te laten reviewen.

Ons devies is: zorg dat jouw project niet alleen bestaat uit louter 'user stories', maar maak altijd heldere documentatie. In een agile omgeving zijn documentatie en borging hiervan, onderdeel van de '*definition of done*'. Bij voortdurende integratie hoort ook het continu op orde houden van je documentatie!

5. Stel bij en lever duidelijke input

Projecten zijn steeds vaker iteratief en kort cyclisch. Dit stimuleert het continu verbeteren van architectuur op basis van gerealiseerde oplossingen en verkregen inzichten. Lever als analist duidelijke input en feedback aan voor de architect: wat werkt wel en wat niet? Je kunt hiervoor de architect direct betrekken bij evaluaties gerelateerd aan het project. Andere projecten kunnen zo van verbeterde architectuur profiteren.

Samenwerking met architect

Conclusie

In complexe projecten is samenwerking met de architect(en) een must. Architectuur is daar een randvoorwaarde voor het behalen van de bedrijfsdoelstelling.

De samenwerking met de architect is niet alleen voor jouw project nuttig. Als professional heb je namelijk ook de verantwoordelijkheid om kennis te delen. Niet opnieuw het wiel uitvinden, maar bestaande initiatieven versterken is daarin heel krachtig.

3. De samenwerking tussen analist en UX designer

De analist verzamelt wat de wensen van de business zijn, structureert ze en legt ze begrijpelijk vast. De UX designer bepaalt hoe de oplossing voor de eindgebruiker werkt en hoe deze eruit ziet.

De analist en de UX designer

Tijdens de voorbereiding van een sprint zorgt de analist ervoor dat de stories aan de bovenkant van de product backlog voldoende niveau hebben, zodat het team de omvang kan inschatten.

In tegenstelling tot de meeste andere rollen in het team heeft de UX designer – net als de analist – vaak een rol in deze voorbereiding. De manier waarop de UX designer de buitenkant van de oplossing ontwerpt, kan immers gevolgen hebben voor de complexiteit en de hoeveelheid werk.

Waar *binnen* de sprint wordt gewerkt aan user stories, zal in de (sprint)voorbereiding vaak worden gewerkt met features. Daarin wordt de oplossing benaderd vanuit een bredere context. De analist breekt samen met de UX designer een feature op in meerdere user stories met bijhorende acceptatie-criteria, zodat deze tijdens een sprint kunnen worden gerealiseerd.

Daarbij gebeurt het dat teams in design sprints in een kort tijdsbestek zaken willen toetsen met business stakeholders en ontwikkelaars om zo nieuwe functionaliteiten of nieuwe proposities te ontdekken of verkennen.

Voor meer afgebakende functionaliteiten of processen in een reguliere heartbeat is het ook wenselijk om een timebox toe te passen. Reserveer daarom tijd waarin je samen met de UX designer een concept kan ontwikkelen voor de oplossing van een story, dat (net) voldoende uitgewerkt is, zodat het team de effort kan inschatten.

Het voorbereiden van een sprint met de UX designer

Tips voor als je als analist de sprint optimaal voorbereiden met de UX designer:

- Baken de scope af in user stories. Bepaal al vooraf – samen met de UX designer – in welke mate de vanuit een feature afgeleide user story impact gaat hebben op de user interface van de oplossing en in hoeverre daarvoor een specifiek UX design voor nodig is. Zorg ervoor dat de user story altijd te herleiden is naar dit specifieke UX design door bijvoorbeeld de gehanteerde naamgeving van een user story terug te laten komen in het UX design.

- Bewaak *scope creep*. Stem met de UX designer af hoe de acceptatie-criteria in een feature of story worden afgedekt in het ontwerp. Hiermee voorkom je dat onnodige functionaliteit en complexiteit wordt toegevoegd, zonder dat daar een legitieme business-wens tegenover staat.

- Gebruik een Kanban board voor features. Een Kanban board geeft het voordeel dat een feature te benaderen is vanuit diverse invalshoeken (o.a. beheer, architectuur, enz), zo ook vanuit het gezichtspunt van een UX designer. Voor de UX designer is een feature de werkeenheid op basis waarvan hij of zij het ontwerp zal maken. Een goed UX design zal immers altijd uitgaan van een afgesloten en complete flow en niet direct van de kleinere sub-processen die je typisch in een user story aantreft.

  De UX designer zal in deze voorbereiding al moeten bepalen wat de impact van een feature zal zijn op de user-interface (front-end) van de oplossing. Wanneer deze geraakt wordt, is het zijn rol om - eventueel met behulp van gebruikerstesten - een ontwerp te bedenken dat op een gebruiksvriendelijke, consistente en technisch realiseerbare manier invulling geeft aan de functionele requirements.

De samenwerking tijdens de sprint

Het komt voor dat ontwikkelaars bepaalde zaken in de realisatie net iets anders willen doen in verband met de complexiteit en/of de hoeveelheid werk. Als analist waak je erover dat nog steeds recht wordt gedaan aan de oorspronkelijke business-wens bij eventuele afwijkingen. De UX designer bewaakt in hoeverre de realisatie in lijn blijft met zijn design voor de story.

Onze praktische tips voor de analist

- Plan een tussentijdse demo. Het is cruciaal dat het team in geval van afwijkingen de discussie voert. De analist kan een rol spelen om de discussie te faciliteren door een tussentijdse demo in te plannen. Het team kan eventuele uitdagingen of afwijkingen tijdens de realisatie van een story dan toelichten in deze demo. Dit voorkomt eventuele verrassingen aan het einde van de sprint.

Tips aan het einde van de sprint

- Laat de UX designer een rol vervullen bij het bedenken en opzetten van de presentatievorm van de sprint review.

- Evalueer aan het einde van de sprint – samen met de UX designer – of de gekozen implementatie van een story nog steeds aansluit bij het integrale ontwerp van de feature.

- Het is ook mogelijk dat een bepaalde implementatie een verrijking is op bestaande standaarden en/of guidelines. Om hergebruik en consistentie te bevorderen dient de UX designer dit te borgen in de documentatie.

- Laat de UX designer daarnaast wijzigingen met grote impact op de front-end zelf presenteren aan de stakeholders. Hij/zij kan zelf het beste uitleggen waarom gekozen is voor een bepaalde implementatie.

Samenwerking met de UX designer - Conclusie

UX design is een vak. We zien vaak dat deze taak niet altijd past in de snelle oplever-cyclus van Scrum (sprints), de UX designer werkt meer op het niveau van features. De analist kan de UX designer faciliteren/helpen om de designtaken rond een user story toch zoveel mogelijk binnen de heartbeat uit te voeren.

4. De samenwerking tussen analist en ontwikkelaar

De analist verzamelt wat de wensen van de business zijn, structureert deze en legt ze begrijpelijk vast. De ontwikkelaar bepaalt hoe hij met software deze wensen vervult.

De analist en de ontwikkelaar

Tijdens de voorbereiding van een sprint zorgt de analist ervoor dat de stories aan de bovenkant van de product backlog voldoende niveau hebben, zodat het Scrum team, met daarin de ontwikkelaar, de omvang kan schatten.

Het is de rol van de analist om in de voorbereiding al rekening te houden met andere systemen en verschillende architecturen. Dit geeft in de refinement een solide basis voor discussie.

Het is de rol van de ontwikkelaar om alvast na te denken hoe de implementatie zal zijn en waar er mogelijke dure consequenties opduiken. Dit geeft in een vroeg stadium de mogelijkheid om andere oplossingsrichtingen te verkennen.

Als de ontwikkelaar aangeeft dat stories te groot zijn voor een sprint, kan de analist de stories in overleg met het team opsplitsen in beter behapbare brokken. Elke story moet klein genoeg zijn voor realisatie in een sprint, maar ook groot genoeg blijven om waarde voor de business op te leveren.

Verder moet elke story net genoeg (*just enough*) informatie bevatten, zodat het team de story kan 'pokeren'. Een spel Progress Poker kan helpen om snel duidelijkheid over de (kwaliteit) van de requirements te krijgen.

Een story mag dus niet te weinig details, maar ook zeker niet te veel details bevatten. Het is *waste* als de analist in de sprint-voorbereiding te veel details uitwerkt. In ons artikel Efficiënte analyse: Just Enough in de praktijk staat meer over het belang van 'just enough' voor projecten.

Het voorbereiden van een sprint met de ontwikkelaar

Wil je als analist de sprint optimaal voorbereiden?
Dan zullen deze tips je helpen:

- Werk met Use Case 2.0: In deze methode kan je een use case in kleine stappen uitwerken totdat de specificatie net helder genoeg is voor jouw team. Het plaatst de user story in de context van de bestaande functionaliteit en dat maakt het herkenbaar voor de ontwikkelaar. De user story vormt dan meteen ook de basis voor systeem-documentatie.

- Geef in de refinement van een user story ruimte voor de ontwikkelaar om te komen met alternatieven voor de realisatie. Bewaak wel dat deze verschillende oplossings-richtingen het doel van de story dienen en er niet onnodig veel aan toevoegen. *Scope creep* ligt op de loer.

- Zorg dat er altijd acceptatie-criteria zijn bij de story, zodat voor iedereen duidelijk is wanneer de story af is.

- Als je de acceptatie-criteria in de given-then-when vorm schrijft, geven ze niet alleen een helder beeld over de user story, maar zijn ze in de meeste gevallen ook direct te gebruiken voor acceptatietesten en unit testen.

De samenwerking tijdens de sprint

Tijdens de sprint vult de analist samen met het team de verdere details van een story in. De analist kan de ontwikkelaar helpen met de structuur van de oplossing. De ontwikkelaar helpt de analist met het signaleren van losse eindjes die bij de realisatie aan het licht komen.

Praktische tips voor de analist

- Doorloop bij de start van de sprint de story nog een keer samen met de ontwikkelaar (en tester). Soms is de refinement weggezakt en moet het geheugen worden opgefrist. Daarnaast is het direct een eerste review van de specificaties en komen onduidelijkheden snel naar voren.

- Leg de specificaties centraal vast. Stuur geen losse mails met specificaties, maar verwijs naar de centrale plaats.

- Als je bestaande documentatie bijwerkt, maak de wijzigingen dan inzichtelijk voor de ontwikkelaar en zorg voor goed versiebeheer. DiVetro heeft werkende oplossingen voor tools zoals Enterprise Architect.

- Bij complexe flows is het belangrijk dat de code goed aansluit op de specificaties. Een *activity diagram* kan dan een goed houvast bieden aan de ontwikkelaar. Complexe interactie tussen meerdere systemen of componenten kan je inzichtelijk maken in bijvoorbeeld een UML sequence diagram.

- Als je met de ontwikkelaar in dezelfde ruimte zit, kan je direct onduidelijkheden opvangen en bespreken, zodat je daarvoor geen aparte sessies hoeft te plannen.

- Het is verleidelijk technische keuzes van de ontwikkelaar vast te leggen in de specificaties, maar daarvoor zijn betere alternatieven. Soms is het goed documenteren van de code voldoende en in andere gevallen kan technisch documentatie zoals bijvoorbeeld *Use Case Realisations* een oplossing bieden.

- Binnen een service-georiënteerde omgeving zal de architect en/of ontwikkelaar een belangrijke rol spelen in het beleggen van verantwoordelijkheden in de componenten. Zeker als het om herbruikbare services gaat, is en blijft de functionele insteek van de analist noodzakelijk. Door services functioneel te documenteren, bijvoorbeeld als “use case” in een aangepaste vorm, weten (potentiële) afnemers wat ze van een service kunnen verwachten.

Tips aan het einde van de sprint

- Aan het einde van de sprint neemt de analist samen met de ontwikkelaar en tester de opgeleverde software door, want er is een grote kans dat het toch net anders of beter is gebouwd dan vooraf bedacht. Een laatste review van de ontwikkelaar op de systeem-documentatie geeft de beste garantie dat deze ook echt aansluit met wat er is gebouwd.

- Ook bij het samenstellen van de release-notes kunnen analist en ontwikkelaar elkaar helpen. Een goed ingericht Scrum bord in combinatie met slim documenteren maakt het zelfs mogelijk om de release-notes te genereren. Over dit specifieke onderwerp verwijzen we graag naar de presentatie die DiVetro hierover heeft verzorgd op het Dreamevent.

- Tijdens de sprint review zal de analist de business context van een oplevering geven, waarna een ontwikkelaar of tester een scenario laat zien. De combinatie van vertalen naar de wereld van de stakeholders en het scenario kunnen aanpassen naar aanleiding van vragen van de stakeholders, geven de beste feedback tijdens zo'n sprint review. Neem je Sprint review dus serieus (zie bonus hoofdstuk).

Samenwerking met de ontwikkelaar

Conclusie

Binnen een IT-project werkt een analist het meest samen met een ontwikkelaar. Daarbij is de analist leidend in het **wat** en de ontwikkelaar in het **hoe**.

Onderlinge kruis-bestuiving is essentieel. Richt dus je ontwerp zodanig in, dat het aansluit op de gebruikte manier van ontwikkelen en sta open voor suggesties van de ontwikkelaar, hij weet niet zelden slimme implementaties te vinden, omdat hij de geheimen van de techniek kent.

Een open en actieve samenwerking zal altijd leiden tot een beter resultaat!

5. De samenwerking tussen analist en tester

De analist verzamelt wat de wensen van de business zijn, structureert deze en legt ze begrijpelijk vast. De tester valideert dat de software deze wensen vervult. Deze samenwerking zorgt voor een gevalideerde (software)oplossing die waarde geeft voor de business.

De analist en de tester

Tijdens de voorbereiding van een sprint zorgt de analist ervoor dat de stories aan de bovenkant van de product backlog voldoende niveau hebben, zodat het Scrum team, inclusief tester, de omvang kan schatten.

Het is de rol van de analist om in de voorbereiding al goed na te denken over de acceptatie-criteria van iedere user story / requirement. Wanneer is iets nou echt af? Wanneer voldoen we

als team nu eigenlijk aan de specifieke wensen van de stakeholder(s). De acceptatie-criteria moeten zorgen voor een zo scherp mogelijke afbakening van de story.

Het is de rol van de tester om alvast na te denken hoe gevalideerd kan worden dat de oplossing ook echt aan deze acceptatie-criteria voldoet. Door in een vroeg stadium te valideren of acceptatie-criteria echt testbaar zijn, voorkomen we een hoop discussie in het vervolg. Het zal ook de eerste keer niet zijn, dat tijdens de refinement de tester missende alternatieven aandraagt of bestaande acceptatie-criteria helpt aan te scherpen.

Bij teams die werken op basis van Specification by Example (SBE), Test-Driven Development (TDD) of Behaviour Driven development (BDD) zijn acceptatie-criteria vaak de basis voor geautomatiseerde testgevallen. Hierbij is de rol van de tester in combinatie met de analist zeer belangrijk om tot zo scherp mogelijke testgevallen te komen.

Het voorbereiden van de sprint samen met de tester

Wil je als analist de sprint samen met de tester optimaal voorbereiden? Dan zullen deze tips je helpen:

- Voorzie iedere story altijd van acceptatie-criteria die ook echt testbaar zijn. Acceptatie-criteria als bijvoorbeeld: “de gebruikersvriendelijkheid gaat omhoog” kunnen lastig te valideren zijn. De wijze van opschrijven kan helpen bij het automatiseren van testgevallen (zie ook vorig hoofdstuk).
- Laat je story voor refinement alvast toetsen door de tester.
- Voorzie complexe flows van een activity diagram. Een activity diagram gaat de tester helpen om snel en simpel tot de diverse testgevallen te komen.

Tips voor tijdens de sprint

Tijdens de sprint vult de analist samen met het team de verdere details van een story in. De analist kan de tester helpen met de opstellen van de testgevallen. De tester helpt de analist met het signaleren van onduidelijke of niet-testbare requirements.

- Betrek de tester bij het opstellen van de requirements. Een tester kijkt op een andere wijze naar de requirements en kan de analist helpen om de zaken scherp op te schrijven.
- Organiseer *three amigo* sessies om in een vroeg stadium tester, ontwikkelaar en de analist bij elkaar te brengen. In de praktijk levert dit vaak goede resultaten in zowel de testbaarheid en de kwaliteit van de code.
- Bied altijd aan om de testgevallen te reviewen. In de review kun je direct controleren of de testen compleet zijn, je requirements niet ambigue zijn en of je geen requirements vergeten bent.
- Leg relaties tussen requirements en testgevallen. Probeer samen met je tester requirements daadwerkelijk te koppelen met testgevallen. Dit kan middels tooling, maar veelal helpt een naamgevingsconventie ook al om snel de relatie te leggen tussen testgevallen en bijbehorende requirements.
- Als je bestaande documentatie bijwerkt, maak de wijzigingen dan inzichtelijk voor de tester en zorg voor goed versiebeheer. DiVetro heeft werkende oplossingen onder meer voor Enterprise Architect.

- Het is verleidelijk om testcondities die de tester gebruikt, vast te leggen in de specificaties, maar daarvoor zijn betere alternatieven. Een praktisch voorbeeld zijn foutmeldingen. Het requirement is veelal dat het systeem een foutmelding zal geven; de specifieke tekst van de melding moet uiteraard logisch zijn, maar zal in 9 van de 10 gevallen niet als “hard” acceptatie-criterium gelden. Probeer samen met het team te komen tot een situatie die voor iedereen werkbaar is. In het geval van foutmeldingen zou een tabel met teksten voor meldingen met een link naar specifieke documentatie wellicht al voldoende kunnen zijn.

- Vergeet niet het gewenste gedrag voor foutsituaties te beschrijven. Sluitende acceptatie-criteria beschrijven niet alleen de happy flow, maar ook wat het systeem moet doen als zaken anders lopen dan gehoopt.

Tips aan het einde van de sprint

Tijdens de sprint review zal het team de resultaten van de sprint aan de stakeholders tonen. Niets is storender voor stakeholders dan een review waarin onrealistische scenario’s en/of data wordt gebruikt. Het gebruik hiervan zal de stakeholder afleiden van de daadwerkelijke werking. En het kan ook een onrealistisch beeld geven van de applicatie.

Stem daarom altijd met de tester af welke set testdata gebruikt zal worden tijdens de demo. Vanuit je expertise als analist kun jij goede realistische scenario’s beschrijven. De tester kan op zijn beurt deze scenario’s omzetten naar een goede consistente set van testdata.

Samenwerking met de tester

Conclusie

Binnen een IT-project werken een analist en tester nauw samen om te zorgen dat de door het team ontwikkelde oplossing daad-werkelijk aansluit bij de gestelde eisen.

Daarbij is de analist leidend in het wat en de tester in valideren dat het ook echt zo werkt.

6. De samenwerking met de beheerder

De analist verzamelt wat de wensen van de business zijn, structureert deze en legt ze begrijpelijk vast. De beheerder waarborgt ten behoeve van continuïteit en kwaliteit, dat de realisatie van deze wensen kunnen worden geïntegreerd in de huidige informatievoorziening. Deze samenwerking zorgt voor een toekomstvaste (software-) oplossing, die blijvend waarde geeft voor de business.

Voor veel agile teams ligt de focus binnen een sprint vaak op het ontwikkelen van nieuwe functionaliteiten en minder op het beheer van een oplossing die al in productie staat. Door deze focus dreigen de beheer-aspecten van de oplossing nog wel eens onder te sneeuwen. Gelukkig maken in een DevOps benadering de beheerders onderdeel uit van het team, waardoor ontwikkel- en beheer-processen dichter bij elkaar komen.

Tijdens de voorbereiding van een sprint zorgt de analist ervoor dat de stories aan de bovenkant van de product backlog voldoende niveau hebben, zodat het development team de omvang kan schatten. Het is de rol van de analist om in de voorbereiding rekening te houden met de impact op andere systemen en architectuur-richtlijnen.

Het is de rol van de beheerder om na te denken hoe de voorgestelde implementatie past binnen de bestaande informatievoorziening. Daarnaast zorgt de beheerder er ook voor dat, indien nodig, ook stories op de backlog komen vanuit een beheer-oogpunt. Hierbij kan bijvoorbeeld gedacht worden aan het schalen van de oplossing naar meer gebruikers of de transitie naar nieuwe hardware of infrastructuur.

Het voorbereiden van de sprint samen met de beheerder

Wil je als analist de sprint samen met de beheerder optimaal voorbereiden? Dan zullen deze tips je helpen:

- Geef user stories over beheer voldoende diepgang. De product backlog wordt door de Product Owner en de analist samengesteld. De stories zijn veelal business value driven. De beheerder kan als stakeholder stories aandragen ten behoeve van het optimaliseren en in stand houden van de informatie-voorziening. In de praktijk sneeuwen deze user stories vaak onder, omdat de Product Owner onvoldoende inzicht heeft in de toegevoegde waarde ervan. Het is de taak van de analist om samen met de beheerder deze stories van voldoende diepgang te voorzien, zodat de Product Owner in staat is om deze stories van de juiste prioriteit te voorzien.

- Aandacht voor foutscenario's. Zorg ervoor dat de fout-scenario's in stories ook worden belicht in de vorm van acceptatie-criteria. De beheerder heeft in zijn analyse dit inzicht nodig in geval van productieverstoringen.

- Gebruik een Kanban bord voor features. Als je zo'n bord inricht, kan de beheerder in een vroegtijdig stadium van het voortbrengingsproces op een abstracter niveau een feature beoordelen op beheer-aspecten en eventueel zijn goedkeuring verlenen. Dit Kanban board is ook nuttig voor andere rollen zoals een UX-designer, architect of Product Owner.

- Organiseer ***four*** *amigos* sessies. Naast het traditionele trio in een three amigos sessie (analist, ontwikkelaar en tester) is het raadzaam om ook de beheerder actief mee te nemen in de voorbespreking van user stories en/of features. Naast gemeenschappelijk begrip en betrokkenheid zorgt dit ervoor dat de beheerder op de hoogte is van wat er op hem afkomt. Een beheerder heeft daarnaast een breder (IT) netwerk in de organisatie en kan de organisatie zodoende tijdig voorbereiden op een wijziging in de informatievoorziening.

Praktische tips voor tijdens de sprint

Tijdens de sprint vult de analist samen met het team de verdere details van een story in. De analist kan de beheerder helpen met de opstellen van de voorlopige releasenotes en het voorbereiden van de transitie. De beheerder helpt de analist met het aanscherpen en vastlegging van de (non-) functionele requirements.

- Betrek de beheerder bij het opstellen van de requirements. Naast een functionele kennisoverdracht helpt dit de beheerder bij het analyseren van productieverstoringen. En in geval van wijzigingen op bestaande documentatie kan de impact beter worden bepaald op de informatievoorziening. De beheerder kan de functionele documentatie gebruiken als naslagwerk en als startpunt om na te gaan of hij te maken heeft met een bug op het systeem of een Request for Change.

- Opstellen releasenotes. De functionele documentatie van een analist kan een prima basis vormen voor de releasenotes die de beheerder in het kader van wijzigingen-beheer opstelt.

- Werk met Use Case 2.0. In deze methode kan je een use case in kleine stappen uitwerken totdat de specificatie helder genoeg is voor het team. Een belangrijke afnemer van de functionele specificatie is beheer. De beheerder kan de analist verzoeken om bij complexere functionaliteiten, de specificaties in meer detail te laten uitwerken. In ons artikel 'Efficiënte analyse: Just Enough in de praktijk' gaan we verder in op het juiste detailniveau van requirements.

- Beschrijf services functioneel en inclusief relaties met andere systemen. In een service-georiënteerde omgeving is het te beheren systeem voor een beheerder mogelijk slechts één aandachtsgebied. Het helpt om de herbruikbare services functioneel te beschrijven en de relaties met andere systemen inzichtelijk te maken. Zodoende heeft de beheerder snel inzicht in de impact van een productieverstoring in de (support)keten.

- Verifieer inrichtingskeuzes. Vanuit de amigos- of refinement-sessie kan een bepaalde inrichtingskeuze zijn gemaakt met impact op beheer. Denk hierbij bijvoorbeeld aan de ontwikkeling van configuratiebestanden en/of stamtabellen. Verifieer de inrichting samen met de beheerder, zodat hij ervaring opdoet in het gebruik. Daarmee voorkom je dat hij geconfronteerd wordt met verrassingen in productie.

Tips aan het einde van de sprint

- Zoals we eerder hebben genoemd in dit e-boek, valideert de analist met de ontwikkelaar en de tester de ontwikkelde software. Neem deze oplevering van de sprint ook door met de beheerder, inclusief de bijbehorende releasenotes en functionele documentatie.

- De beheerder bereidt de verdere transitie naar productie verder voor. In dit stadium zal de beheerder zorg dragen voor de communicatie richting de gebruikersorganisatie. Eenmaal in productie is beheer het eerste aanspreekpunt voor de gebruikers-organisatie. De gemelde incidenten, wijzigingsverzoeken en overige opmerkingen van gebruikers kunnen weer leiden tot nieuwe items op de product backlog. Zorg er als analist voor dat deze zaken ook als zodanig door de Product Owner worden geprioriteerd en behandeld. Hiermee voorkom je dat gemelde incidenten, wijzigings-verzoeken en overige opmerkingen een eigen leven gaan leiden en op een aparte lijst naast de product backlog belanden.

Samenwerking met de beheerder

Conclusie

Bij het voortbrengingsproces is het van belang dat de analist en beheerder nauw samenwerken ten behoeve van een passende oplossing binnen de gestelde kaders en huidige informatievoorziening.

7. De samenwerking tussen analisten onderling

In dit hoofdstuk staat de samenwerking tussen analisten onderling centraal.

Voorbeeldscenario's

Laten we beginnen met een stukje context. Gezien de gemiddelde samenstelling van een Scrum team kunnen we ons voorstellen dat niet iedereen een beeld heeft bij de samenwerking van analisten onderling.

Waar er doorgaans in een ontwikkelteam slechts één analist in het team zit, zijn er wel degelijk scenario's waarin een team meerdere analisten heeft of waarin samenwerking vereist is met analisten uit een ander team. Denk bijvoorbeeld aan:

- de periode waarin een uitstromende analist zijn werkzaamheden overdraagt aan een nieuw teamlid;
- een periode waarin het team een sterke afhankelijkheid heeft met functionaliteiten die opgeleverd moeten worden door een ander team;
- een periode waarin extra capaciteit gevraagd is om nieuwe Features voor te bereiden voor bijvoorbeeld een aankomend PI event (SAFe);
- een periode waarin een team wordt gesplitst in twee teams om de toenemende vraag te kunnen ondersteunen.

Drie praktische tips voor een betere samenwerking tussen analisten

1. Verdeel de verantwoordelijkheid

Als we kijken naar scenario's waarin meerdere analisten samen verantwoordelijk zijn voor bijvoorbeeld één feature, één applicatie of één team, dan is het in de praktijk vaak lastig om de werkverdeling helder en transparant te houden.

Daar waar je samen verantwoordelijk bent, is het makkelijk om ook samen zaken op te gaan pakken. Dit heeft uiteraard als voordeel dat je veel inhoudelijk kunt sparren en dat je gezamenlijk tot een gedeelde oplossing kunt komen. Het nadeel is echter dat je de capaciteit op deze manier niet optimaal benut.

Onze ervaring leert dat het in zo'n scenario loont om de verantwoordelijkheden scherp te verdelen en goede afspraken te maken over het eigenaarschap van features:

- Wees transparant in het eigenaarschap. Door voor iedereen helder te maken welke analist het primaire aanspreekpunt voor een story of feature is, voorkom je verwarring en weet iedereen wie het eigenaarschap heeft van de functionaliteit.

- Beperk de focus. Door het eigenaarschap en de daarbij horende verantwoordelijkheden te verdelen voorkom je dat meerdere analisten zich vastbijten in hetzelfde onderwerp en dat daarmee capaciteit en focus verloren gaat. Uiteraard is het nog steeds logisch om daar waar nodig actief kennis te delen en te sparren over eventuele uitdagingen. Uiteraard helpt een WIP limit ook prima om de focus te behouden.

- Zorg voor benaderbare documentatie. Gezien het feit dat slechts één analist het primaire aanspreekpunt is, is het belangrijk om de beschikbare informatie (documentatie) toegankelijk op te zetten en te houden voor andere analisten / geïnteresseerden. Een teamwiki is vaak handig, maar niet als je informatie actief wilt delen met mensen buiten je team.

Benut de capaciteit door een heldere verdeling van verantwoordelijkheden

2. Stem werkwijze af en deel kennis

Een veelvoorkomende valkuil voor agile teams is dat zij de focus te veel op het eigen team hebben gericht. Een risico hiervan is dat niet, of onvoldoende, aansluiting gezocht wordt bij een gedeelde werkwijze voor het vastleggen van requirements / documentatie. Zeker in een omgeving waarin analisten actief kennis moeten delen, loont het de moeite om hier effort in te stoppen. Kennis delen kan bijvoorbeeld via analisten-gildes waar men behaalde resultaten en best practices kan delen.

Een gedeelde werkwijze heeft de volgende voordelen:

- Verhoging van de flexibiliteit. Door de werkwijzen op elkaar af te stemmen wordt de onderlinge inwisselbaarheid groter en kunnen de analisten elkaar makkelijker vervangen en ondersteunen. Hierdoor wordt de flexibiliteit van de organisatie en de value stream vergroot.

- Verhoging van de continuïteit. Een gedeelde werkwijze zorgt voor grotere continuïteit voor het team en maakt communicatie met de andere teams makkelijker.

3. Communiceer!

Zoals bij iedere vorm van samenwerking is communicatie het belangrijkste punt om rekening mee te houden. Zonder actieve communicatie zullen mensen zich verschuilen achter procedures en dreigt “shaming en blaming” indien zaken niet lopen zoals verwacht. Zoek daarom de communicatie actief op en plan zo nodig een afstemmingsoverleg tussen betrokken analisten met bijvoorbeeld de gedeelde Product Owner(s).

Binnen deze actieve communicatie zal “planning” vaak ook een meer prominente rol gaan spelen. Zeker indien je als team afhankelijk bent van werkzaamheden van een ander team, is het essentieel om transparant en helder inzicht te geven in de planningen van de diverse teams.

Uiteraard is dit een belangrijk onderdeel in de voorbereiding van het PI event. Na het PI event zal dit in een Scrum-of-Scrums ook afgedekt worden. Het kan echter geen kwaad om hier ook gewoon buiten dit afstemmingsmoment transparant in te zijn.

- Deel planningen. De meest primaire vorm is uiteraard om je Scrum board op een toegankelijke plaats op te hangen zodat iedereen kan zien waar aan gewerkt wordt / zal worden. Indien je met behulp van een elektronisch board werkt is het simpel om je board openbaar te maken voor andere gebruikers.
- Visualiseer afhankelijkheden. Naast het simpel delen van je planning loont het de moeite om afhankelijkheden met andere teams inzichtelijk te maken. Dit kan bijvoorbeeld door een simpele kleur- / notatie-conventie op je papieren board, maar ook door de zaken elektronisch te koppelen, bijvoorbeeld via “dependency” relaties.

Samenwerking onderling

Conclusie

Een optimale samenwerking tussen analisten bereik je door een heldere scheiding van de verantwoordelijkheden, afstemming van de werkwijze, het communiceren van planningen en visualiseren van afhankelijkheden.

8. Neem je sprint review serieus

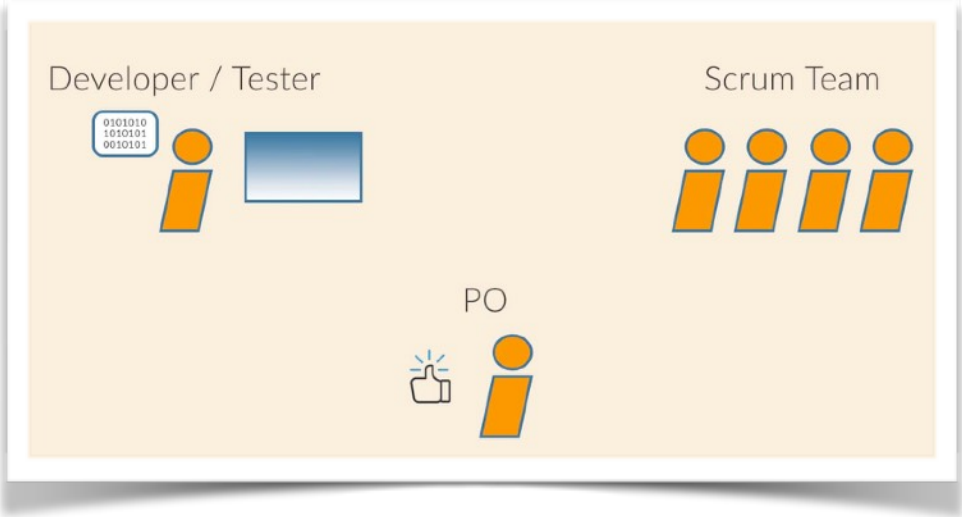
Als bonus een hoofdstuk met handige tips voor een betere sprint review.

De sprint review is één van de standaard-onderdelen van een Scrum project, die bij elke sprint opnieuw wordt uitgevoerd. Het is de moeite waard om het optimale resultaat te halen uit deze review, maar dat lukt niet altijd.

We zien in onze dagelijkse praktijk te vaak dat de sprint review wordt beschouwd als een formele afsluiting van de sprint. De Product Owner keurt de stories goed en het Scrum team is blij met de goede afloop van de sprint.

Stakeholders zijn meestal niet uitgenodigd of haken na enkele sprints af, veelal doordat de demo te technisch is of doordat de demo wordt gegeven met onjuiste of niet representatieve testdata. Maar:

Een sprint review uit de praktijk



Het doel van een sprint review is juist om feedback van stakeholders te verkrijgen. Denk daarom goed over welke vorm en werkwijze voor het project het beste resultaat oplevert.

Het doel van de sprint review

Tijdens een sprint review inspecteren we de afgelopen sprint en verzamelen we feedback van de stakeholders. De belangrijkste aandachtspunten tijdens een sprint review zijn:

- Tonen wat er bereikt is (live demo);
- Verzamelen van feedback;
- Bevorderen van onderlinge samenwerking;
- Indien nodig aanpassen van de backlog.

Sprint reviews zijn veelal informeel van aard om de feedback en samenwerking te bevorderen. Dit is niet het moment voor de Product Owner om stories goed te keuren. Stories die tijdens de sprint review aan bod komen, zijn vooraf immers al aan de Product Owner gepresenteerd.

De juiste mensen

Om zinvolle feedback te kunnen verzamelen is het belangrijk dat de juiste mensen bij de sprint review aanwezig zijn. Bepaal daarom welke stakeholders feedback kunnen geven en nodig ze tijdig uit. Denk daarbij aan:

- Product Owner;
- Senior users/ambassadeurs;
- Business verantwoordelijken;
- Scrum team;
- Operatie (indien niet vertegenwoordigd in het Scrum team);
- Implementatie.

Wijs een verantwoordelijke aan voor de uitnodiging en het actief monitoren van de opkomst. Als een belangrijke stakeholder meerdere keren niet aanwezig is, zoek dan persoonlijk contact en probeer belemmeringen weg te nemen.

Mogelijk is de dag of het tijdstip niet gunstig. Of misschien is de demo een keer inhoudelijk niet interessant geweest. Meld vooraf wanneer een demo mogelijk minder belangrijk is voor een genodigde. Dan kan de genodigde nog altijd zelf kiezen of hij al dan niet aanwezig zal zijn.

Goede voorbereiding

Een goede sprint review vereist ook een gedegen voorbereiding. Als pas enkele uren van tevoren over de sprint review wordt nagedacht, dan zal deze niet optimaal zijn. Door op tijd over de review na te denken zal de kwaliteit verbeteren. Hierbij stellen we de volgende richtlijnen voor:

- Reserveer aan het begin van de sprint daadwerkelijk tijd voor de voorbereiding. Maak hiervoor bijvoorbeeld een algemene taak aan op het Scrumboard voor iedere sprint. Of maak per story die je in de review wil laten laten zien een “review” taak.
- Start op tijd met de voorbereiding: zodra duidelijk wordt dat een story daadwerkelijk zal worden opgeleverd in de sprint, niet de dag voor de review.
- Bespreek de scenario’s die in de review worden getoond met de betrokken teamleden.
- Verzamel tijdig aansprekende en productie(-like) data.
- Stel de Product Owner tijdig op de hoogte.
- Stel omgevingen veilig voor het tijdslot van de review.

Schets altijd de context van een story

In een sprint review ligt de nadruk terecht vaak grotendeels of uitsluitend op een live demo van software. Stakeholders waarvan feedback wordt verwacht, zijn echter niet altijd bij de sprint betrokken geweest en kunnen geholpen worden met een korte situatieschets of inleiding van de story. Daarom:

- Schets altijd de context per story zodat de stakeholder weet waar hij feedback op moet geven en wat er nog gaat komen en dus geen onderwerp van discussie is.
- Beschrijf de feature/epic waar de story bij hoort als context.
- Beschrijf de actuele status van de feature/epic. Benoem stories die al zijn afgerond en die nog op de backlog staan om onnodige discussie te voorkomen.
- Beschrijf het doel en de resultaten in business termen.

Deze schets/inleiding is goed vorm te geven met behulp van allerlei presentatietechnieken. Dit zal vaak een Powerpoint-presentatie zijn, maar filmpjes of andere presentatievormen zijn uiteraard ook mogelijk. Enkele algemene aandachtspunten:

- Markeer de plaats in deze presentatie waar de live-demo zelf gaat komen, bijvoorbeeld met een vast symbool.
- Gebruik de taal van de stakeholders.
- Gebruik de juiste huisstijl van de organisatie.
- Besteed tijd en aandacht aan details.
- Zorg voor een goede verhouding tussen tekst en plaatjes.
- Gebruik herkenbare plaatjes en symbolen van het bedrijf.

Je kan de presentatie ook gebruiken om achteraf afwezigen te informeren. Zorg in dat geval voor voldoende tekst en voeg enkele screenshots van de live demo toe.

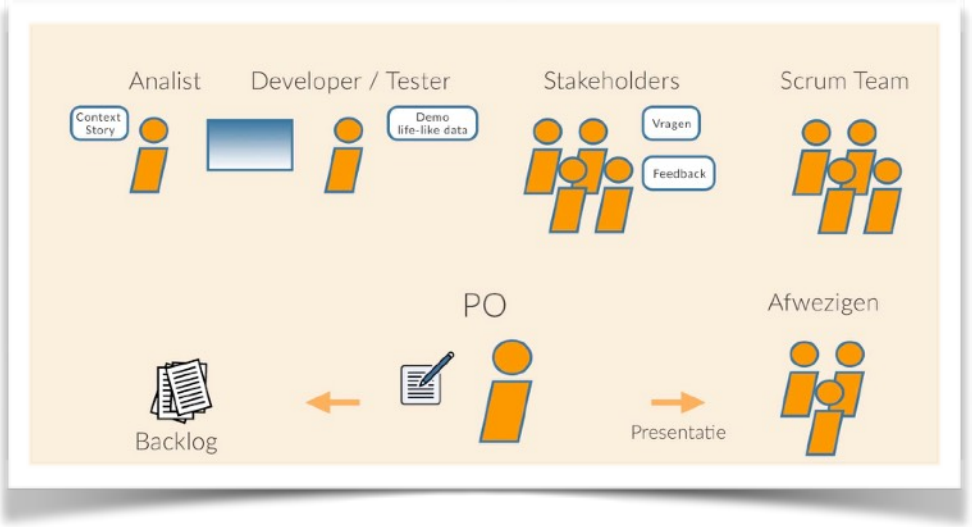
De uitvoering

Begin de sprint review met een korte intro, bijvoorbeeld een korte agenda in de presentatie. Vertel kort over problemen die het team heeft overwonnen en belangrijke gebeurtenissen als releases.

Behandel daarna de diverse stories:

- Vertel zoals hiervoor beschreven eerst de context van de story. De spreker (vaak een analist) moet de stakeholders kennen, hun taal spreken en eventuele vragen kunnen beantwoorden.
- Toon vervolgens indien mogelijk een live demo. De bediening en toelichting wordt gegeven door iemand die handig is met de knoppen en weet hoe hij testdata beschikbaar kan maken. Dit is vaak een developer of tester.
- Zorg voor een vloeiende overgang tussen presentatie en live demo, bijvoorbeeld door een monitor met twee ingangen.
- Maak vooraf afspraken over wie feedback noteert. Dit is bij voorkeur de Product Owner.

Hoe het er dan uit zal zien



Na afloop wordt de feedback door de Product Owner en het SCRUM team besproken. De product backlog wordt zo nodig aangepast, aangevuld of opnieuw geprioriteerd.

Conclusie

Goede sprint reviews vragen voorbereiding en tijd. Dit zal zich snel terugbetalen doordat stakeholders beter betrokken worden en ze daardoor vaker bij de sprint review aanwezig zijn, waarbij ze gerichtere feedback geven en waardoor de kwaliteit van de backlog ook weer verbetert. Al deze zaken zullen bijdragen aan het succes van het project.

DANKWOORD

Dit e-book is een bundeling van de kennis en ervaring van onze professionals op het gebied van analyse.

Wij bedanken hierbij graag al onze opdrachtgevers en collega's die bewust en onbewust hebben bijgedragen aan de totstandkoming van onze inzichten en dit e-book. Zonder hen was deze publicatie niet mogelijk geweest.

DiVetro
Digitale Vooruitgang

OVER DE AUTEURS

Dennis Geluk, senior informatie-analist en partner bij DiVetro:

Mijn uitdaging in het werk is om samen met klanten tot de best mogelijke oplossing te komen voor een probleem. Dit klinkt eenvoudig, maar hoe vaak zie je niet dat we middels een technische invulling iets proberen op te lossen zonder echt te kijken naar het daadwerkelijke probleem? Oplossingen moeten ontstaan uit samenwerkende mensen die ondersteund worden door techniek en een proces en niet primair uit de techniek of het proces. De oplossing ligt vaak dichterbij dan je zelf denkt!

Marco Balvers, senior informatie-analist bij DiVetro:

Dienstverlening begint bij een goed idee. Ik verdiep mij in de achterliggende gedachten en werk dit door middel van discussies en workshops uit tot een concreet ontwerp. Het achterhalen van de vraag achter de vraag is daarbij vaak de kritieke succesfactor. Mijn uitdaging is het komen tot een breedgedragen oplossing door eenduidige communicatie en menselijke interactie.

DiVetro
Digitale Vooruitgang

Marco van Dop, senior informatie-analist bij DiVetro:

Het leukste aspect van het ontwerpvak is dat het mij de mogelijkheid geeft om me volledig te kunnen verplaatsen in de wereld van de klant. Het is vervolgens de uitdaging om met een frisse kijk op de situatie samen met de klant te komen tot de best passende oplossingen. Een gestructureerd en begrijpelijk ontwerp vormt de basis voor een eindproduct dat voldoet aan de verwachtingen en echt bruikbaar is in de organisatie.

Ronald Koenis, senior informatie-analist bij DiVetro:

Ik haal energie uit het vertalen van business vraagstukken naar optimale IT-oplossingen. Het begint met inzicht krijgen in complexe domeinen en daar structuur in aanbrengen. Die complexiteit moet vervolgens worden teruggebracht naar een oplossing die voor de business te begrijpen is en tegelijkertijd voldoende specifiek is, zodat IT er mee aan de slag kan. En dat op een manier doen die zowel optimaal is voor de business als voor IT, daar ligt mijn uitdaging.

DiVetro
Digitale Vooruitgang

Selmar van Egmond, senior informatie-analist bij DiVetro:

Voor mij begint een goed project met een gedegen analyse en het definiëren van een stip op de horizon. Een goede analyse gaat terug naar de werkelijke oorzaken van het probleem, zodat een effectieve oplossing kan worden gezocht. De stip op de horizon zorgt voor een doelgerichte aanpak voor alle betrokkenen. Het vervolgens bedenken en neerzetten van een werkende oplossing komt tot stand met een open houding en interactie. Hiervoor creëer ik een omgeving waarbij iedereen in zijn kracht kan staan.

Stanley Lachman, senior informatie-analist bij DiVetro:

Een team is succesvol als ieder teamlid zich verantwoordelijk voelt voor de te behalen doelstellingen, open staat voor elkaar en er sprake is van behulpzaamheid en heldere communicatie. Mijn voldoening als analist haal ik dan ook uit het leveren van een bijdrage aan een gezamenlijk waardevol resultaat. Het veranderingsproces dat met de totstandkoming gepaard gaat, intrigeert me en houdt me gepassioneerd voor dit vakgebied.

DiVetro
Digitale Vooruitgang

Tijn van Wegen, senior informatie-analist bij DiVetro:

Nog geen 100 jaar geleden was een tocht naar het volgende dorp een wereldreis. Nu kunnen organisaties globaal samenwerken en activiteiten integreren in ketenbrede processen die in snel tempo kunnen wijzigen. Een goede informatievoorziening is randvoorwaardelijk voor een organisatie om te kunnen functioneren in deze dynamiek. Mijn uitdaging is om (complexe) proces- en IT-vraagstukken te beantwoorden met creatieve en betrouwbare oplossingen.

DiVetro
Digitale Vooruitgang

OVER DIVETRO

DiVetro is een onafhankelijk strategisch adviesbureau voor IT-gerelateerde vraagstukken. De wereld positiever maken door te werken aan de digitale vooruitgang bij onze opdrachtgevers, dát is onze ambitie.

Wij zijn vooral actief op projecten bij de (semi-)overheid, in de gezondheidszorg en in het onderwijs. Onze consultants hebben ruime ervaring op hun vakgebied.

Onze werkwijze is onafhankelijk en transparant. Wij geven concreet en eerlijk advies en we houden datgene wat we doen simpel en begrijpelijk voor alle betrokkenen. Zo zorgen wij ervoor dat iedereen de veranderingen binnen de organisatie van onze opdrachtgevers begrijpt en ondersteunt.

Kwaliteit en ondernemerschap staan centraal in onze bedrijfsfilosofie. Wij nemen verantwoordelijkheid, stimuleren dat bij anderen en scheppen zo een omgeving waarin iedereen van elkaar leert en samenwerkt aan een gezamenlijk doel. Hier plukken onze collega's en onze opdrachtgevers de vruchten van.

Bezoek ons op divetro.nl.