

Aandacht voor requirements in een agile werkomgeving

Vandaag de dag zijn veel mensen geïnteresseerd in agile productontwikkeling. Veel mensen werken in een agile omgeving of zouden dat graag willen. Sommige mensen denken agile en anderen denken dat ze agile zijn. In veel gevallen wordt Scrum als raamwerk gebruikt om complexe producten op een agile manier tot stand te brengen. Scrum is krachtig door zijn eenvoud en laat nog erg veel ruimte open om het ontwikkelwerk in te vullen.

door Sven van der Zee

Die vrijheid sluit mooi aan op het agile idee dat mensen in interactie met elkaar een belangrijkere bijdrage leveren aan het resultaat dan een gedefinieerde werkwijze. In de praktijk vinden mensen het echter vaak lastig om - naast het verrichten van de werkzaamheden - te blijven nadenken over hoe ze te werk zullen gaan en hoe ze hun werk kunnen verbeteren. Je ziet dan dat er klakkeloos wordt vertrouwd op voorgeschreven of populaire technieken zonder dat de essentie ervan wordt doorgrond. Je kunt dan ook zeggen dat de professionaliteit en expertise tekort schieten om de ruimte die door Scrum wordt geboden op een goede manier in te vullen. Een vakgebied dat bij het invullen van die ruimte goed van pas komt is requirements engineering. Wie deze expertise in een agile team inbrengt, kan variëren. Volgens Scrum heeft de Product Owner de eindverantwoordelijkheid voor het vullen en onderhouden van de Product Backlog. De Product Backlog is de verzameling van requirements voor een Scrum team. Het zorgen voor een goede Product Backlog doet de Product Owner niet alleen. Dat gebeurt - als het goed is - in interactie met stakeholders en het ontwikkelteam. De mate waarin en de manier waarop een Product Owner in de praktijk geholpen moet worden bij dit requirementswerk verschilt enorm. Dat hangt onder andere af van alle kennis en vaardigheden van de persoon die de zware rol van Product Owner vervult. De Product Owner wordt vaak ondersteund door analisten en ontwikkelteamleden met requirements engineering expertise. Deze expertise schiet ook daar wel eens te kort waardoor de kwaliteit van de agile requirements te wensen over laat en het ontwikkelproces stroef loopt of dat het gerealiseerde product niet voldoet. Het is de kunst de essentie van het vakgebied requirements

engineering te vertalen naar een agile praktijk. Overeenkomsten tussen toepassing van requirements engineering in een traditionele werkomgeving en een agile werkomgeving schetsen de essentie van dit vakgebied. Stilstaan bij de cruciale verschillen helpt bij de vertaling naar de praktijk.

Overeenkomsten in doel en denkwijzen

Een klassieke valkuil in requirements engineering is om te denken dat het alleen gaat over de opgeschreven requirements. Echter, het ultieme doel van requirements engineering is om een gemeenschappelijk beeld en een gedeeld begrip van de vraagstelling te bewerkstelligen tussen alle betrokkenen. Die vraagstelling omvat het geheel van verwachtingen, behoeften, wensen en eisen die leven bij de stakeholders. Komen tot zo'n gemeenschappelijk beeld kan gebaat zijn bij een bepaalde documentatievorm zoals een traditionele Software Requirements Specification (SRS) of een agile user story. Maar de ervaring leert dat het minstens even belangrijk is om over de vraagstelling te praten, zodat het begrip hiervan echt wordt gedeeld. Face-to-face interactie wordt in agile kringen omarmd als meest effectieve communicatievorm en krijgt daardoor een extra accent ten opzichte van het documenteren. De weg die wordt bewandeld om het doel van requirements engineering te bereiken, mag dan verschillen. Het ultieme doel blijft gelijk. In alle gevallen dit doel voor ogen houden helpt om vanuit de essentie van requirements engineering bezig te zijn en niet te verdwalen in de toepassing van technieken. Daarnaast helpt het om je bewust te zijn van de generieke denkwijzen binnen requirements engineering los van de specifieke technieken en terminologie die daarbij worden

gehanteerd. De denkwijzen van requirements engineering in een traditionele werkomgeving of een agile werkomgeving komen in hoge mate overeen. Voorbeelden van generieke denkwijzen in requirements engineering zijn: van probleem naar oplossing; van globaal naar detail; diverse invalshoeken dragen bij aan de kwaliteit van requirements; technische en financiële consequenties werken door in afbakening en prioriteit van de vraagstelling.

Verschillen in timing en technieken

Het verschil tussen requirements engineering in een traditionele werkomgeving en een agile werkomgeving zit vooral in een andere timing en een voorkeur voor andere technieken.

Andere timing

Het is niet agile om alle requirements in één keer tot in detail te specificeren. In de Sprints van Scrum pakt het ontwikkelteam steeds een beperkte set van requirements op in plaats van alle requirements tegelijk. In een agile team moet je er samen voor zorgen dat steeds vlak voordat de nieuwe versie van het werkende product gemaakt gaat worden alle betrokkenen snappen wat de bedoeling is van de volgende stap (just enough, just in time requirements). De details van de vraagstelling komen steeds per Sprint aan de orde. Door in een kort tijdsbestek een behapbaar deel van de behoeften van de stakeholders te vertalen naar een werkende oplossing is er een snelle terugkoppeling. Deze herhaaldelijke terugkoppeling maakt het goed mogelijk dat er wordt bijgestuurd in de requirements en het requirementsproces. Doordat het bepalen van requirements in de tijd wordt uitgesmeerd en een terugkerende activiteit is, is het afstemmen op de consequenties ook meer een doorlopend proces en niet iets dat alleen in een initiële projectfase plaatsvindt.

Andere technieken

Agile requirementstechnieken ondersteunen meer het idee van gezamenlijke inspanning rondom afzonderlijke requirements dan het traditionele idee van een specialistische rol die werkt aan uitgebreide requirementsdocumenten. Agile teams leggen requirements liever vast op een laagdrempelige manier met information radiators die de interactie stimuleren. Bekende voorbeelden hiervan zijn: user stories, story maps, vision board. Traditioneel wordt minimaal onderscheid gemaakt in functionele en niet-functionele requirements. Dit vertaalt zich meestal in aparte hoofdstukken in de traditionele requirementsdocumenten. Het hanteren van een dergelijk onderscheid draagt vooral bij aan de volledigheid van de requirementsverzameling. In een agile omgeving worden functionele en niet-functionele aspecten gevangen door afzonderlijke requirements (veelal losse user stories) gezamenlijk te analyseren. Bijvoorbeeld aan de hand van de zeven productdimensies van Ellen Gottesdiener (www.discovertododeliver.com). Dit leidt in ieder geval tot een beter begrip bij de betrokkenen, leidt vaak tot nieuwe

requirements (opsplitsen van user stories) en leidt soms tot aanvullende details in de vorm van acceptatiecriteria per requirement. Traditioneel gaat de afstemming van de gehele vraagstelling op consequenties meer in termen van technische haalbaarheid en betaalbaarheid. In een agile omgeving wordt de nadruk vooral gelegd op de toegevoegde waarde en de technische afhankelijkheden en risico's van afzonderlijke requirements.

Het is een kunst om goede input voor het ontwikkelproces te krijgen. Deze expertise hoeft niet bij één specialist te liggen.

To engineer or not to engineer

In veel agile raamwerken wordt het aanleveren van requirements als een autonoom proces voorgesteld: stakeholders komen zelf met requirements op de proppen. In de praktijk komt hier echter veel creativiteit en vakmanschap bij kijken. Het is een kunst om goede input voor het ontwikkelproces te krijgen. Dit vakmanschap werd en wordt requirements engineering genoemd. Deze expertise hoeft niet bij één specialist te liggen. Als agile team wil ik dat er voldoende requirements engineering expertise aanwezig is, zodat requirements de aandacht krijgen die ze verdienen.



Sven van der Zee

Sven is trainer, coach en adviseur bij inspearit | cibit academy. Hij heeft als bedrijfskundige 21 jaar werkervaring op het snijvlak van business en IT. Zowel bij profit als non-profit organisaties heeft Sven teams en

individuen geholpen om verbeteringen in processen en resultaten te bereiken. Daartoe beheerst en hanteert hij diverse methoden en technieken. Sven heeft in de loop der jaren al veel mensen geholpen meer uit zichzelf en uit hun organisatie te halen op basis van het vakgebied requirements engineering en op basis van agile ideeën en raamwerken. Zijn ervaringen deelt hij graag met anderen.