

Om in te kunnen spelen op de vraag vanuit de markt wordt het voor veel organisaties steeds belangrijker om nauwkeurige schattingen van softwareprojecten af te kunnen geven. Klanten vragen vaak een vaste prijs en opleverdatum. Om geen grote risico's te lopen en niet uit de markt geprijsd te worden, is een juiste inschatting van de projectomvang van groot belang. Diverse schattingsmethoden pretenderen de oplossing te zijn voor deze problematiek maar blijken in de praktijk vaak erg complex of aan subjectiviteit onderhevig. Is schatten met smart use cases wel de oplossing voor al onze problemen?

Schatten met smart Use Cases

Welkom alternatief met aantal sterke punten

Waarom levert het schatten van softwareprojecten eigenlijk zoveel problemen op? Veranderende requirements tijdens het project, inzet van nieuwe onbeproefde technologieën, optimisme van ontwikkelaars, hopen dat een project beter gaat dan het vorige en reverse planning zijn slechts een aantal aanwijsbare oorzaken van foutieve schattingen. Beproefde schattingsmethoden zoals FunctiePunten Analyse (FPA), Lines of Code en expert-schattingen leveren vaak niet het gewenste resultaat op. Het schatten met smart use cases is daarom een welkom alternatief met een aantal sterke punten.

Wat zijn smart use cases?

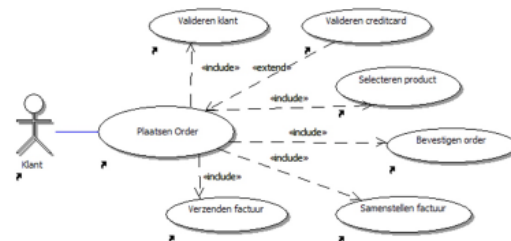
Smart use cases hebben niets te maken met Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdgebonden (S.M.A.R.T.) zoals je wellicht zou kunnen denken. Een smart use case is afkomstig van een Agile-stroom, genaamd Accelerated Delivery Platform (ADP). De use cases bij de meeste ontwikkelmethoden zijn meestal gedefinieerd op User Goal Level, ook wel Sea Level genoemd in sommige literatuur. Hierbij vertegenwoordigt een use case een gebruikersdoel welke verder wordt uitgediept in de specificaties.

Smart use cases gaan een niveau dieper; subfunction level ofwel 'Fish level'. De verschillende subfunctionaliteiten van de hoofd use case worden visueel weergegeven in een diagram. Het totaal van smart use cases geeft een gedetailleerd beeld van de werking, omvang en scope van het systeem. Deze smart use cases kunnen al in de beginfase van het project gedefinieerd worden zonder dat de exacte

werking ervan volledig bekend is. De use case 'Plaatsen order' is in figuur 1 uitgewerkt op het niveau van smart use cases.

Smart use case punten

Als alle smart use cases gedefinieerd zijn wordt een sessie georganiseerd waarin het hele projectteam, en bij voorkeur ook de opdrachtgever, bij elkaar komt om de use cases te beoordelen. Deze schattingsmethode wordt ook wel Smart Estimation genoemd. De waardering van de smart use cases loopt van 1 tot 10 punten. Alle aanwezigen, van ontwikkelaar tot tester, kent een aantal punten toe aan elke smart use case naargelang de complexiteit. Het is ook mogelijk dat een functionaliteit nog volledig onbekend is, bijvoorbeeld als er met een ander, nog onbekend systeem gekoppeld wordt. Het wordt aangeraden om dergelijke onzekerheden met 10 punten te beoordelen. Na verloop van tijd is het raadzaam om richtlijnen op te stellen om vast te leggen hoe veel voorkomende functionaliteit geteld moet worden. Dit is beter voor de reproduceerbaarheid en betrouwbaarheid van schattingen.



Figuur 1 - voorbeeld van een use case op subfunction level.



Maik ten Eikelder
is informatieanalist op de afdeling MAS bij Capgemini.

Na de individuele beoordelingen worden de resultaten met elkaar vergeleken. Wanneer er sprake is van afwijkingen tussen de gemaakte inschattingen dan wordt dit besproken. Het is belangrijk dat alle facetten van een stukje functionaliteit worden belicht. Een use case kan functioneel eenvoudig zijn, maar door de technische mensen als complex beoordeeld worden. Uiteindelijk ontstaat er overeenstemming over de moeilijkheidsgraad van elke smart use case en is elke use case gewaardeerd met een aantal punten. In vergelijking met veel andere schattingsmethoden wordt de schatting door het gehele team gedragen en begrijpt zelfs de opdrachtgever de geschatte complexiteit van alle afzonderlijke onderdelen.

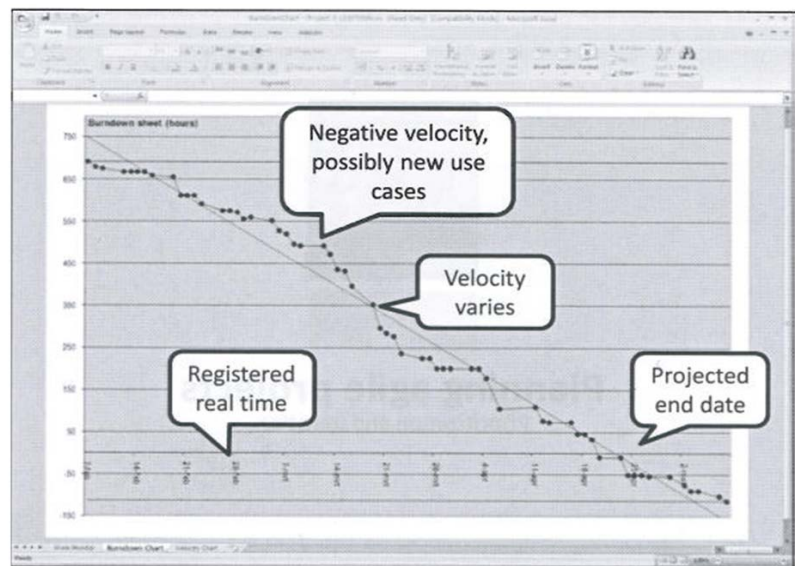
Schatten en plannen

Uit meetgegevens van eerdere projecten kan een organisatie het gemiddelde aantal smart use case punten bepalen dat binnen een bepaalde tijd wordt gerealiseerd. Door deze 'snelheid' met het totale aantal use case punten van het project te vermenigvuldigen wordt de totale doorlooptijd van het project bepaald. Na de eerste iteratie kan de snelheid van het project bepaald worden (men weet immers hoeveel smart use case punten in een bepaalde tijd zijn gerealiseerd) en kan de planning eventueel bijgesteld worden.

Voor elke iteratie mag de opdrachtgever voor een x aantal punten (afhankelijk van de lengte van de iteratie) aan use cases kiezen om te realiseren. De opdrachtgever, die betrokken is geweest in het voortraject en de opgestelde specificaties begrijpt, kiest de meest essentiële functionaliteiten eerst. Het kiezen van een minder belangrijke maar complexe smart use case zou immers weinig ruimte over laten voor het realiseren van andere (wellicht eenvoudigere) noodzakelijke functionaliteiten binnen de iteratie. De valkuil om teveel tijd te besteden aan niet essentiële stokpaardjes wordt zo vermeden. Tevens worden de belangrijkste onderdelen van het project in een vroeg stadium opgeleverd. Als nieuwe smart use cases worden geïdentificeerd tijdens het project dan worden deze beoordeeld en op zijn vroegst in de volgende iteratie gerealiseerd.

Waarom deze methode?

Schatten met smart use cases is geen 'silver bullet' voor al onze schattingsproblemen, maar er zijn zeker voordelen in vergelijking met andere schattingsmethoden. Veel methoden leveren als resultaat een omvang van een systeem in een bepaalde eenheid zoals functiepunten. Waarom niet schatten op de manier waarop we onze projecten doen? Aangezien de meeste organisaties ontwerpen, bouwen en testen met use cases ligt het voor de hand om use cases ook te gebruiken in de schattingsmethodiek. Dit resulteert in een reëlere afstemming van schattingen en werkelijke uitvoering van een software project. Een ander voordeel is dat schattingen niet gemaakt worden door



commerciële afgezanten met als belangrijkste doel scoren of door grijze analisten in stoffige kamertjes. Nee, de schatting is een gezamenlijke inspanning van het ontwikkelteam waarin alle facetten worden meegenomen. Tevens draagt het schatten met smart use cases bij aan de kennisopbouw van het project en is het niet alleen maar een onderdeelje van het voortraject dat later weer 'vergeten' wordt.

Het resterende werk van een lopend project met smart use cases is uitstekend te monitoren middels een dashboard. Door de snelheid te berekenen met behulp van de resterende use case punten op een bepaalde tijd kan eenvoudig een overzicht (velocity diagram) gegenereerd worden met daarbij de geschatte eindtijd. Hierdoor kunnen vroegtijdig beslissingen genomen worden die de voortgang beïnvloeden zoals extra mankracht aan het project toevoegen of de scope veranderen. Er wordt geschat en opgeleverd in smart use cases, dus is het mogelijk om goede sturing aan het project te geven.

De methode schatten met smart use cases hoeft natuurlijk niet meteen de bestaande methodieken te vervangen, maar zou ook naast bijvoorbeeld Funtiepunctanalyse gebruikt kunnen worden. Smart use cases zijn trouwens nauw verwant aan het Agile proces maar zijn natuurlijk ook toepasbaar op andere ontwikkelmethoden zoals de RUP lifecycle. Het aantal Agile aanhangers wordt wereldwijd steeds groter en ook vanuit de klanten wordt de vraag naar Agile steeds vaker gesignaleerd. De ontwikkeling verleidt steeds meer softwareorganisaties om gehoor te geven aan deze roep om een Agile methode te implementeren. Aangezien de schattingsmethodiek van projecten nauw verwant is met de projectaanpak is het schatten met smart use cases een interessante methode die het proberen waard is. De eerste ervaringen opgedaan binnen mijn eigen organisatie zijn in ieder geval veel belovend. «

Figuur 2 - Voorbeeld van een velocity diagram, resterende use cases in de tijd

Het ligt voor de hand ook te rekenen in use cases als ze normaal ook worden gebruikt