

Een van de nieuwste ontwikkelingen binnen agile is Kanban. Een concept afkomstig van Lean Manufacturing¹. Dit concept is door o.a. David Anderson^{2,3} vertaald naar software ontwikkeling. Inmiddels is deze manier van werken door veel meer mensen bij diverse bedrijven met goede resultaten toegepast. Kanban is in staat een project of bedrijf flexibel en transparant te maken.

Kanban

Nieuwe ontwikkeling binnen Agile

De vakanties zijn voorbij en de gebruikelijke ochtend- en avondrituelen vinden weer plaats op de Nederlandse snelwegen. Steeds langer wordende files, continu stoppen en optrekken. Het is (in het westen van Nederland) moeilijk om aan te ontsnappen. Het probleem is eigenlijk vrij eenvoudig: de hoeveelheid te verwerken eenheden (auto's) is groter dan de capaciteit van het systeem (de snelweg) waardoor er een harmonica effect⁴ ontstaat en alles langzamer gaat.

Bij de ingang van de keizerlijke tuin in Tokyo krijgt elke bezoeker een plastic kaart die zij bij het verlaten van de tuin weer afgeven aan de bewaking. De bedoeling van deze constructie is dat het een heel eenvoudige manier is om het aantal bezoekers in het park te beperken. Over het algemeen zijn er voldoende kaarten aanwezig en kan iedere bezoeker gewoon doorlopen. Op drukke dagen kan het gebeuren dat een nieuwe bezoeker moet wachten tot een andere bezoeker het park heeft verlaten. Iedere bezoeker krijgt hierdoor een optimale ervaring van zijn bezoek, terwijl er toch zo veel mogelijk mensen in het park kunnen.

Twee verhalen die op het eerste gezicht niets te maken hebben met softwareontwikkeling. Toch is het onderliggende probleem van beide situaties wel van toepassing op softwareontwikkeling. Of we het willen toegeven of niet; we hebben vaak een probleem met 'flow' in projecten. Kanban helpt deze op te lossen door de flow zichtbaar te maken en deze vervolgens te verbeteren.

Lean

Lean manufacturing en lean engineering zijn gebaseerd op het werk van Henry Ford en William Edwards Deming. Het doel van Lean is een lerende organisatie op te zetten die zich richt op het reduceren van verspilling. Deze verspilling doet zich op vele fronten voor: overbezetting, variaties

in aan- en afvoer, voorraden, niet-waarde toevoegende activiteiten, etc.

Lean is vooral bekend geworden door Toyota die tijdens de wederopbouw na de 2e wereldoorlog gedwongen was optimaal gebruik te maken van de beschikbare middelen, inclusief grondgebruik. Het bedrijf heeft zich derhalve helemaal gericht op het zo effectief mogelijk inrichten van de productie en enkel datgene te doen wat noodzakelijk is en waarde oplevert.⁵

Push/pull

Een van de middelen die Toyota gebruikt is een Kanban systeem. Doelstelling is om het gehele proces te laten sturen op basis van de klantvraag (pull systeem). Dus in plaats van te beslissen hoeveel je wilt of maximaal kunt produceren, wat in de traditionele productie systemen gebeurt (push systeem), bepaalt de klant hoeveel wordt geproduceerd. Een mooi voorbeeld van een pull systeem is het 'two bin' principe, waarbij voorraad continu wordt aangevuld zodra de huidige voorraad half op is⁶. In plaats van steeds de voorraad bij te vullen gebeurt dit nu pas als dit ook werkelijk wordt gevraagd door de productie.

De traditionele manier van werken is een push systeem. Het voordeel hiervan is dat het centraal kan worden aangestuurd. Het product wordt verplaatst zodra het klaar is. Het nadeel van deze manier van produceren is dat het kan leiden tot suboptimalisatie van het systeem door enkel subsystemen te optimaliseren. De totale effectiviteit van het systeem komt hierdoor in het geding. Daarnaast leidt het tot grote (tussen)voorraden.

Een pull systeem is klantgestuurd en elk proces communiceert met het vorige en het volgende proces. Het gehele productieproces is op deze manier met elkaar gesynchroniseerd. Je zou kunnen zeggen dat elk proces weer de klant is van het

Olav Maassen

is Chief Engineer bij QNH en
secretaris van AgileHolland
(olav.maassen@qnh.nl)

vorige proces. Er is geen centrale aansturing meer nodig, terwijl het proces wel visueel te volgen is. Er wordt alleen gemaakt wat nu noodzakelijk is en de voorraden blijven klein. Door deze kleine voorraden is het ook mogelijk sneller te reageren op veranderingen. Een eenvoudig en betrouwbaar systeem.

Uiteraard is ook het pull systeem niet zonder nadelen. Dit systeem vraagt veel discipline van de medewerkers om alleen op de vraag van de klant te reageren, dit kan ook een klantproces zijn. Er kan ook geen schaduwsysteem naast het pull concept worden gebruikt.

Kanban kaarten

Door een limiet te plaatsen op het maximaal aantal taken (Kanban kaarten) dat tegelijkertijd kan worden uitgevoerd krijgen we een systeem gelijk aan dat van de keizerlijke tuinen. Er is een limiet voor zowel het totale systeem als voor een individuele kolom of groepering van kolommen. Deze kolommen komen overeen met de verschillende activiteiten die in een proces plaatsvinden.

Doordat er minder werk in het systeem zit, is de snelheid waarmee de taken door het systeem bewegen hoger, waarmee een grotere doorstroom wordt gecreëerd en de totale capaciteit van het team stijgt.

Om verschillen in snelheid en wachttijden op te lossen zijn er extra queues en buffers nodig. Op deze queues en buffers plaatsen we ook een Kanban limiet om ervoor te zorgen dat we geen grote voorraden gaan opbouwen. Deze buffers zijn er enkel en alleen om kleine variaties in doorlooptijd op te vangen. Het streven is om het systeem optimaal te laten verlopen en daarnaast iedereen zo effectief mogelijk in te zetten.

Zodra een taak is afgerond is deze beschikbaar voor de volgende stap in het proces. Deze taak gaat er niet vanzelf heen, maar pas als de volgende stap er om vraagt. Deze volgende stap is de 'klant' van de activiteit waar de taak nu zit. Pas als deze taak is verschoven is er weer ruimte in de huidige kolom en kan een volgende taak worden gevraagd aan het vorige proces. Op deze wijze trekt het systeem functionaliteiten door het systeem heen op basis van de klantvraag.

De Kanban kaarten (taken) zijn niet alleen taken, die moeten worden uitgevoerd. Je kunt het vergelijken met hoe we met geldbiljetten omgaan, dit is ook niet enkel papier. Geld heeft een waarde omdat we dat met elkaar afgesproken hebben. We kunnen zelfs zo ver gaan dat geld tijd is en dat het geld dient als ruilmiddel van tijd. Dit werkt alleen doordat er een limiet zit op hoeveel geld in omloop is en doordat niet iedereen zo maar geld kan produceren.

Op dezelfde manier is een Kanban kaart slechts beperkt voorradig. Zo lang er niets is opgeleverd

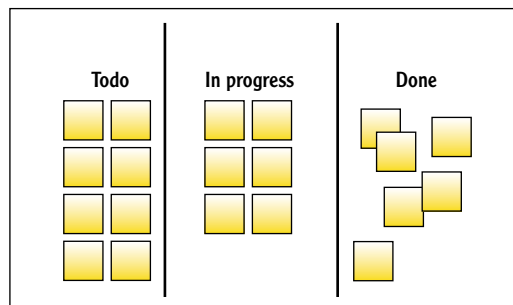
kan er ook geen nieuwe taak bijkomen. Hierdoor ligt de focus op het oplossen van de huidige problemen. Daarnaast zorgt kwaliteitsverbetering ervoor dat taken niet meerdere keren moeten worden opgepakt.

Kanban board

Veel ontwikkelaars zijn tegenwoordig wel bekend met een taskboard waarop een team bijhoudt welke taken zij nog moeten doen, met welke zij bezig zijn en welke taken klaar zijn. In Scrum heet dit ook wel een Scrum board. Hoe zet je nu een taskboard om naar een Kanban systeem?

Enkel het bijhouden van taken zorgt niet voor een Kanban systeem. Wat we willen bereiken is een Kanban systeem, waarbij functionaliteit middels een optimale flow door het systeem wordt getrokken en er zo min mogelijk voorraden in het systeem zijn. In een productieproces zijn voorraden makkelijk te zien, bij software is dat lastiger. In principe is al het werk dat in het systeem zit en waar een ontwikkelaar nu niet direct aan werkt voorraad. Hoe meer werk er tegelijkertijd in het systeem zit, hoe langzamer het geheel gaat. Wat we dus willen doen is de hoeveelheid werk (work-in-progress oftewel WIP) beperken.

Voor het opzetten van een eenvoudig Kanban board hebben we 3 stappen nodig. Deze stappen hoeven niet perse in de beschreven volgorde plaats te vinden. Vaak beginnen ze wel in volgorde en past het team vervolgens het Kanban board aan aan de nieuwste inzichten en wensen.



Figuur 1

Eerst gaan we kijken welke activiteiten het team onderneemt om tot een gereed product te komen. De eerste versie is vaak verdeeld in 3 activiteiten of stadia: todo, in progress en done (zie figuur 1). Later verschuift dit dan naar: todo, analyse, design, development, test, done. En weer later zullen de behoeften verder zijn gewijzigd en zullen kolommen worden gesplitst of samengevoegd.

Als de initiële kolommen zijn bepaald is het tijd om te kijken hoeveel capaciteit beschikbaar is om deze werkzaamheden uit te voeren. Idealiter werkt iedereen altijd aan slechts 1 taak tegelijk en pas als deze klaar is begin je aan de volgende taak.

**Elk proces
is de klant
van het
vorige proces**

Uit onderzoek is gebleken dat met elke keer dat we van taak wisselen we 15% minder productief worden⁷. Wat dan als we niet verder kunnen met een taak? In dat geval begin je aan een tweede taak terwijl je wacht tot de blokkade van je eerste taak is opgelost. Onder geen geval start je ook een derde taak. Dit zou er voor zorgen dat de ontwikkelaar minder productief wordt, omdat hij zijn aandacht moet verdelen en gaat er net als in de file een harmonica effect optreden.

Vervolgens gaan we limieten zetten op de kolommen. De vuistregel is om tussen de 1 en 2 taken per beschikbare ontwikkelaar aan te houden. Dus als we 4 ontwikkelaars hebben die aan taken kunnen werken, ligt de Kanban limiet op die kolom tussen de 4 en de 8. Omdat niet iedereen altijd aan 2 taken zal werken, is in dit geval de meest waarschijnlijke limiet 6. Dus, 2 mensen kunnen aan 2 taken tegelijk werken, de andere 2 kunnen dan geen extra taken meer oppakken (zie figuur 2).

De te nemen stappen voor het opzetten van een Kanban board of het omzetten van een taskboard naar een Kanban board zijn dus:

Bepaal welke activiteiten plaats vinden en plaats deze in verschillende kolommen

Kijk wat de verwerkingscapaciteit is per kolom of per groepering van kolommen en plaats limieten op de kolommen om de WIP te beperken en pull te creëren.

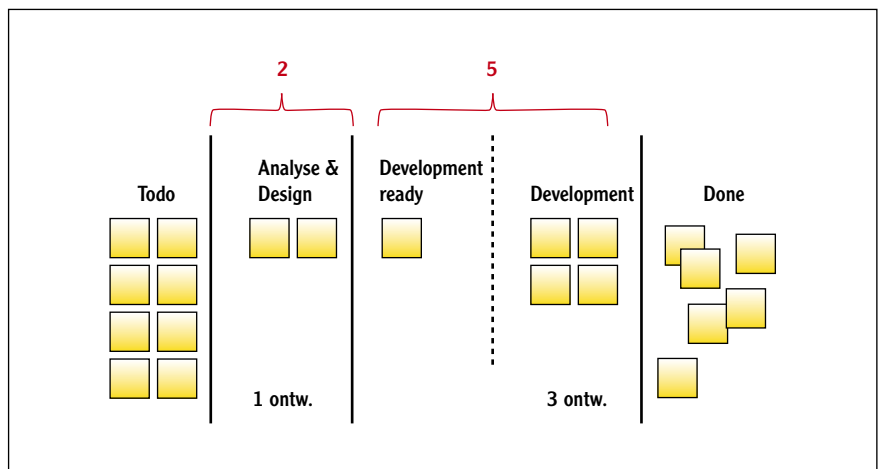
Plaats queues en buffers om kleine variaties in snelheid op te vangen.

Het visuele systeem met zijn limieten laat nu heel snel zien waar zich problemen voordoen. Als de taken te groot zijn, zal de flow met horten en stoten gaan. Een te grote taak zal de andere taken in de processtappen daarvoor gaan ophouden, terwijl de processtappen erna leeg lopen. Kleinere taken gaan veel sneller door het systeem. Grote features opsplitsen in meerdere kleine zorgt dat het geheel sneller geïmplementeerd is.

Het Kanban proces legt een grote focus op kwaliteit. Taken die het team meerdere keren moet uitvoeren zijn verspilling en dat willen we zo veel mogelijk beperken. Door de kwaliteitsfocus bereikt het team een hogere throughput (features gaan sneller door het systeem) en een hogere output (meer opgeleverde features per tijd). Het team zal het proces continue monitoren en kijken waar zij het verder kunnen verbeteren.

Voordelen

Kanban helpt om een ontwikkelproces natuurlijker te laten verlopen. Met Kanban kan een project zonder iteraties gaan werken omdat zij continu features kunnen opleveren. Het team is veel beter in staat om zichzelf te sturen en de klant kan meer op hoofdlijnen sturen. Kanban teams zijn productiever, werken beter samen en hebben een hogere focus op kwaliteit.



Veel teams merken dat het opsplitsen van functionaliteit zodat het in iteraties past lastig is. Zoals gezegd binnen Kanban is het niet nodig om met iteraties te werken. De enige voorwaarde is dat de taken allemaal ongeveer even groot zijn, niet te groot en niet te klein.

Verder kan Kanban zijn waarde bewijzen bij teams waar specialisten zitten die een bottleneck zouden kunnen vormen. Kanban maakt dit zichtbaar en laat snel zien waar deze zich bevinden en helpen je deze maximaal te benutten en de rest van het systeem er op aan te passen, zodat het geheel zo effectief mogelijk is.

Conclusie

Het invoeren van Kanban zal leiden tot een cultuurverandering waarbij het team meer vertrouwen en meer zelfbeschikking zal krijgen, beter zal samenwerken en een (nog) hogere focus op kwaliteit zal leggen. De scope van een release kan nog laat worden gewijzigd, evenals de prioriteit. Het team raakt geen tijd kwijt aan het maken van inschattingen en het zal regelmatig opleveren. «

Referenties

- 1 http://nl.wikipedia.org/wiki/Lean_manufacturing
- 2 <http://www.agilemanagement.net/>
- 3 <http://www.moduscooperandi.com/>
- 4 [http://nl.wikipedia.org/wiki/File_\(verkeer\)](http://nl.wikipedia.org/wiki/File_(verkeer))
- 5 The Toyota Way, Jeffrey K. Liker
- 6 http://nl.wikipedia.org/wiki/Two_Bin
- 7 Slack, Tom de Marco

Figuur 2