Een vergelijkende studie tussen een modulaire, individuele workflow en een traditionele, teamsgebonden workflow bij een Agile framework binnen software development en de invloed hiervan op developers

Onderzoeksvoorstel Bachelorproef 2020-2021

Nick Lersberghe[1]

Samenvatting
Agile is een framework dat in het laatste decennium aan een grote opmars bezig is. Meer en meer bedrijven stappen af van een inflexibel watervalsysteem ten voordele van het flexibele en dynamische Agile framework. Het ene framework is echter het andere niet en verschillende implementaties hebben unieke voor- en nadelen. Dit werk tracht een verband te vinden tussen de gekozen implementatie en zowel de voortgang van het project als het welzijn van de developer. Er zal gekeken worden welke aspecten van de implementaties een effect hebben hierop en hoe deze mogelijks gecombineerd kunnen worden teneinde een betere Agile-implementatie.

Sleutelwoorden
Onderzoeksdomein. IT-projectmanagement — Agile — Software development

Co-promotor
Jocelijn Geurts[2] (Cronos Group, Axudo)

Contact: [1] nicklersberghe@gmail.com; [2] jocelijn@axudo.bejocelijn@axudo.be;

Inhoudsopgave

1. Introductie

Agile is een framework dat in het laatste decennium aan een grote opmars bezig is. Meer en meer bedrijven stappen af van een inflexibel watervalsysteem ten voordele van het flexibele en dynamische Agile framework. Momenteel behoort het tot de standaarden van de industrie. Binnen software development elimineert het enkele van de meest courante problemen die verantwoordelijk zijn voor het falen van veel software projecten.

Het valt te bemerken dat er een grote variatie bestaat in hoe de principes van Agile exact worden toegepast. Frameworks zoals Scrum en Kanban omvatten dezelfde principes met een andere uitwerking. In dit werk zal er geprobeerd worden te achterhalen wat de voor- en nadelen van 2 dergelijke workflows zijn en hoe deze gecombineerd kunnen worden teneinde een beter projectverloop.

De vragen die in dit werk onderzocht zullen worden zijn:

- Welk effect heeft het werken in team aan een feature[1] op de vooruitgang van het project?
- Welk effect heeft het werken in team op het welzijn van de developer (stress, etc.)?
- Hoe kunnen beide workflows gecombineerd worden om het beste van beiden te bekomen en hoe vertaalt zich dat praktisch naar een framework?

2. Stand van zaken

2.1 Principes

De principes waar Agile op rust zijn relatief simpel. Deze werden in 2001 besproken en uitgeschreven door een groep van 17 developers (Beck & et al., 2001). De voor dit werk relevante principes zijn (geparafraseerd en vertaald):

- Omarm veranderende requirements
- Lever regelmatig werkende software

[1] Wanneer er naar 'werken in team' wordt verwezen zelfs al is de developer lid van een team, wordt hiermee altijd bedoeld dat er samen aan een feature of onderdeel wordt gewerkt, tenzij anders aangegeven.

- Een constant tempo moet voor onbepaalde tijd vol te houden zijn
- Werkende software is een maat voor voortgang

"Verwelkom verandering in eisen, ook laat in de ontwikkeling. Agile-processen benutten verandering als een competitief voordeel voor de klant"

(Beck & et al., 2001). De klant is in staat om de processen van projecten te beheersen aan de hand van on-site interactie en requirements geven de huidige noden van de eindgebruikers waarheidsgetrouw weer (Kumar & Bhatia, 2012). Dit wordt gezien als het grootste voordeel van Agile ten opzichte van een klassiek watervalsysteem. Frequente communicatie met de opdrachtgever ('De klant') leidt tot een constante instroom van feedback. In combinatie met een ander voordeel ('Lever regelmatig werkende software', zie verder) is de analist beter in staat de wensen van de klant (of herzieningen hiervan) en de interpretatie van het ontwikkelingsteam ('De developers') gelijk te stellen (of hier zo dicht mogelijk bij te komen). Een bijkomend voordeel is hier dat er een constante herziening van het product is, wat zorgt dat foutopsporing sneller en gemakkelijk gebeurt (Imreh, Raisinghani e.a., 2011).

"Lever regelmatig werkende software op. Liefst iedere paar weken, hooguit iedere paar maanden"

(Beck & et al., 2001). Eén van de essenties van Agile werken, is het iteratief en incrementeel werken. In de praktijk vertaalt dit zich naar het werken in sprints, waarbij er een vaste, korte periode een voorafgeplande hoeveelheid werk verricht wordt, waarna er bekeken wordt hoe dit verliep en wat er aangepast moet worden. In combinatie met het vorige punt ('Omarm veranderende requirements') kunnen veranderende requirements makkelijk geïntegreerd worden. Er gebeurt immers regelmatig een herziening van het product, want bij elke sprint is er een testfase.

"Agile processen bevorderen duurzame ontwikkeling. De opdrachtgevers, ontwikkelaars en gebruikers moeten een constant tempo eeuwig kunnen volhouden"

(Beck & et al., 2001). Hoewel het mogelijks het simpelste principe is, is het tevens één van de belangrijkste. Een tempo dat comfortabel vol te houden is, leidt tot een betere werkomgeving en logischerwijs ook een beter product. Bij Agile zijn de teams waarin gewerkt wordt zelfsturend. De overhead is beperkt tot een stand-up waarbij eerder naar voortgang wordt gekeken. Organisatorische problemen worden tijdens een retrospective (aan het einde van de sprint) besproken en eventueel verbetert op een manier waar de developers zelf mee akkoord gaan, wat de developers gemotiveerd houdt.

"Werkende software is de belangrijkste maat voor voortgang"

(Beck & et al., 2001). Een goede maat hebben is een vereiste om voortgang te kunnen meten. Waar een watervalmethode dit moeilijk maakt, zal bij Agile het iteratief en incrementeel werken leiden tot meer werkende stukken software. Elke feature wordt apart behandeld en aan het einde van de sprint is het de bedoeling dat deze ook afgewerkt zijn. In een ideale situatie (wat echter zelden voorkomt) is er na elke sprint een werkend product waarbij de features minstens minimaal afgewerkt zijn en bij volgende sprints verder aan gewerkt zal worden.

2.2 Toepassing

Het is logisch dat de implementatie en toepassing van deze principes anders zullen zijn afhankelijk van de grootte van het bedrijf en het team zelf. Uit de principes valt op dat het gebruik van Agile te vergelijken valt met change management. Zelfs buiten software development is het een bruikbaar systeem om met verandering om te gaan. "The concept of agile working has been adopted by many organizations which have realized that their hierarchical structures and lengthy decision-making processes are no longer fit for purpose in a world of complex and continuous change" (Franklin, 2014). Hier valt uit af te leiden dat kleinere bedrijven sneller geneigd zullen zijn om Agile te gebruiken, daar deze een minder strikte hiërarchie hebben. Uit Salo en Abrahamsson (2008) valt op dat het gebruik van Agile toeneemt met een toenemende bedrijfsgrootte. Deze toont een verandering in gedachtengoed. Waar vroeger Agile vooral voor kleinere bedrijven en projecten was, is het inmiddels een gevestigde waarde geworden. Ook binnen de zogenaamde 'Fortune 500'-bedrijven, waaronder bv. Alphabet Inc., moederbedrijf van Google LLC, wordt hiervan gebruik gemaakt.

2.3 Implementaties

Zoals onder 'Principes' te bemerken valt, Agile is geen uniform framework. De exacte toepassing van deze principes wordt vastgelegd in aparte frameworks. Hiervan zal er slechts 1 worden besproken, daar het doel van dit werk is om de principes zelf onder de loep te nemen ongeacht welk framework er gebruikt wordt, maar het wel nuttig kan zijn om een concrete implementatie van deze principes te bekijken. Er werd hiervoor gekozen voor Scrum, daar dit het meest courant is (Sutherland & Schwaber, 2007). Scrum is een Agile framework met een relatief simpele structuur. Er zijn drie rollen vastgelegd: de Product Owner, het team en de Scrummaster. De Product Owner is de klant. Deze persoon is tevens beheerder van de lijst van taken ('De product backlog'), bepaalt wat in welke volgorde moet gebeuren. Het team is verantwoordelijk voor het afleveren van het product. Deze groep is multidisciplinair en verantwoordelijk voor alle stappen in het ontwikkelingsproces, gaande van analyse tot de testing. De Scrummaster leidt het team doorheen het Scrumproces. Deze persoon voorziet eventuele trainingen en organiseert de vergaderingen. Hij fungeert als contactpersoon tussen het team en externen, maar is geen project manager. De rituelen die bij Scrum gebruikt worden, zijn o.a. de daily stand-up, sprint planning en retrospective. De daily stand-up is een dagelijkse, korte vergadering waar men al rechtstaand

binnen 10 minuten de to-do, doing en done van elk lid overloopt en eventuele problemen uitlegt en om hulp vraagt waar nodig. Aan het begin van elke sprint gebeurt er een sprint planning, waar het verloop van de komende sprint wordt uitgepland en taken verdeeld. Aan het eind van de sprint gebeurt er een retrospective, waar wordt teruggeblikt op de afgelopen sprint, wat er al dan niet verkeerd is gelopen en hoe dit verbeterd kan worden.

Veel bedrijven gebruiken echter geen vast framework en hanteren een eigen implementatie van de Agile principes. Dit kan zijn omdat het type project het niet toelaat, het team niet de juiste grootte of vorm heeft of de requirements duidelijk en onveranderlijk zijn (Näkki e.a., 2011). Het is dus noodzakelijk dat een bedrijf alle parameters in gedachten houdt, zoals de bedrijfscultuur, het type product en hoe change management gebeurt. Projecten kunnen falen doordat het gekozen systeem niet geoptimaliseerd is voor de developers.

3. Methodologie

3.1 Verzameling data

De data zal verzameld worden door middel van regelmatige enquête. Door de verschillende aard van de twee te onderzoeken workflows en verschillen in project scope, is het noodzakelijk om een correcte schaal te kiezen. Om dit te bewerkstelligen, is het genoeg om naar de structuur van Agile projecten te kijken. Het is een zekerheid dat er bij alle projecten in sprints zal gewerkt worden. Er zal dus per sprint en per week een enquête afgenomen worden die pijlt naar de voortgang van het project en het welzijn van de developer. Er zullen verschillende aspecten bekeken worden als variabelen om af te kunnen leiden of deze al dan niet een effect hebben op beiden. Ten slotte zal er gekeken worden welke aspecten een positieve invloed hebben en welke combinaties er mogelijk zijn. De enquête zal worden afgenomen via een Google Form waarbij er voor de voortgang en het welzijn een 5-punt Likertschaal zal gebruikt worden, waarbij een score van 3 neutraal is (voortgang is zoals verwacht, welzijn is normaal), 1 zeer slecht is (sterk achter op schema, zeer veel stress) en 5 zeer goed (sterk voor op schema, welzijn is zeer goed). Tevens zal er extra gegevens gevraagd worden, zoals natuurlijk het type workflow (individueel - teamgebonden), bedrijfscultuur en mogelijke externe factoren die een invloed kunnen hebben op beiden (teamlid die wegvalt, persoonlijke problemen, etc.). Het gebruik van een Likertschaal zorgt ervoor dat de subjectiviteit van 'welzijn' en 'voortgang' toch enigzins objectief wordt, doordat ze vergeleken worden met de baseline van de developer zelf.

De enquête zal bij een groot aantal developers worden afgenomen, zodat er genoeg data aanwezig is om elke variabele te kunnen isoleren bij de verwerking. Er wordt gericht op 10 bedrijven met een gemiddelde teamgrootte[2] van 5 developers, waardoor de steekproefgrootte rond 50 respondenten ligt.

De te extra onderzoeken variabelen zijn[3]:

- Grootte van het team waarin samengewerkt wordt
- Sprintduur in weken
- Veranderlijkheid van requirements

3.2 Verwerking data

Het verwerken van de data zal neerkomen op het berekenen van de correlatie tussen de variabelen. Uit de dataset zullen er verschillende clusters worden verdeeld, zodat er steeds slechts één onafhankelijke variabele zal zijn. De verwerking zal gebeuren in R of Python. Hieruit zal een correlatie-coëfficient uit worden berekend. De interpretatie zal simpel zijn. Een correlatie boven een bepaalde threshold (± 0.7) zal een sterke correlatie aantonen. Bij een coëfficient tussen ± 0.3 en ± 0.7 zal er sprake zijn van een matige correlatie. Hierbij kan gekeken worden of meerdere variabelen samen een sterke correlatie bekomen.

4. Verwachte resultaten

Er wordt verwacht dat een individuele workflow een betere voortgang betekent, maar een lagere score voor het welzijn van de developer zal hebben. De grootte van het team zal omgekeerd evenredig zijn met de voortgang en het welzijn, m.a.w. grotere teams developers zullen meer stress ervaren en minder voortgang maken op het project. Een kortere sprintduur zal meer stress veroorzaken maar tevens voor veel voortgang zorgen. De mate van veranderlijkheid van requirements zal omgekeerd evenredig zijn met zowel de voortgang als het welzijn. Een verwachte dataset werd geplot in figuren 1 en 2. Hierbij werd voor de eenvoud enkel de voortgang en het welzijn bekeken. Andere variabelen zullen meegerekend worden, maar werden niet visueel voorgesteld.

Volgende figuren zijn louter een visuele voorstelling van een mogelijke dataset. Het aantal punten die samenvallen worden niet speciaal weergegeven. De data van developers die individueel werken werd voorgesteld in figuur 1, voor zij die in team werken is dit in figuur 2. Hier werd de voortgang geplot in functie van het welzijn. De assen stellen de waarden voor, waar 3 een neutrale baseline is, 1 zeer slecht en 5 zeer goed is. Een kleurencode werd gebruikt om te punten visueel een score te geven, waarin rood een slechte situatie voorstelt, blauw een neutrale en groen een goede situatie.

5. Verwachte conclusies

Er wordt verwacht dat een individuele workflow voor voortgang zorgt ten koste van het welzijn van de developer. Tevens zou het werken in grote teams nefast zijn

[2] Grootte van het team waarin samen aan een feature gewerkt wordt

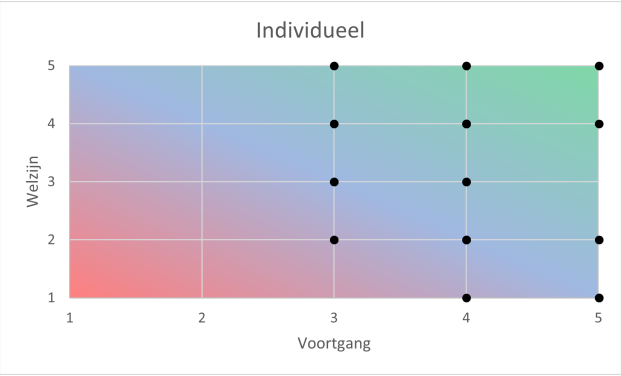
[3] Naast de vergelijking tussen de individuele en teamsgebonden workflow. De lijst is niet limitatief, noch zullen ze noodzakelijk allemaal gebruikt worden.

voor zowel de voortgang als het welzijn. De sprintduur zal omgekeerd evenredig zijn met de voortgang maar recht evenredig met het welzijn. De veranderlijkheid van requirements zal op zich weinig directe invloed hebben op zowel voortgang als welzijn.

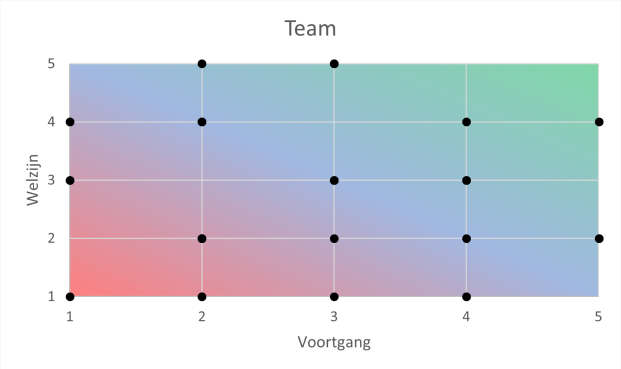
Er wordt tevens verwacht dat zowel de individuele workflow als de teamsgebonden workflow sterk onderhevig zijn aan externe factoren. Er zal echter te concluderen vallen dat er met een combinatie van beide implementaties, waar er in kleine teams van 2 of 3 mensen wordt gewerkt, een balans tussen beide kan bekomen worden. Hierbij kan er een sprintduur van 2 weken worden gehanteerd. De mate van veranderlijkheid van requirements zal echter meestal afhankelijk zijn van de klant en niet de implementatie van een Agile framework.

Op basis van de verwachte data en de verwachte conclusies, zou er een framework kunnen voorgesteld worden waar elementen van Scrum in verwerkt worden, in combinatie met elementen van andere frameworks waar het werken in kleine groepen van 2 of 3 developers aangeraden wordt. Hierbij kan gedacht worden aan praktijken zoals Peer Programming[4] die bij frameworks zoals Extreme Programming (XP) te vinden valt.

[4]Praktijk waar 2 developers regelmatig elkaars werk doornemen en controleren teneinde zaken als het schrijven van duidelijke code



Figuur 1. Voortgang/welzijn-grafiek individueel



Figuur 2. Voortgang/welzijn-grafiek team

Referenties

Beck, K. & et al. (2001). Principles behind the Agile Manifesto. https://agilemanifesto.org/principles.html

Franklin, M. (2014, maart 1). Agile Change Management. Kogan Page. https://www.ebook.de/de/product/21031007/melanie_franklin_agile_change_management.html

Imreh, R., Raisinghani, M. e.a. (2011). Impact of agile software development on quality within information technology organizations. Journal of Emerging Trends in Computing and Information Sciences, 2(10), 460–475.

Kumar, G. & Bhatia, P. K. (2012). Impact of agile methodology on software development process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), 46–50.

Näkki, P., Koskela, K. & Pikkarainen, M. (2011). Practical model for user-driven innovation in agile software development. 2011 17th International Conference on Concurrent Enterprising, 1–8.

Salo, O. & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. IET Softw., 2, 58–64.

Sutherland, J. & Schwaber, K. (2007). The scrum papers. Nuts, Bolts and Origins of an Agile Process.