

SCRUM WEGWIJZER

Een kompas voor de bewuste reiziger



Gunther Verheyen

Scrum Wegwijzer – Een kompas voor de bewuste reiziger

Andere uitgaven bij Van Haren Publishing

Van Haren Publishing (VHP) is gespecialiseerd in uitgaven over Best Practices, methodes en standaarden op het gebied van de volgende domeinen:

- IT en IT-management;
- Enterprise-architectuur;
- Projectmanagement, en
- Businessmanagement.

Deze uitgaven zijn beschikbaar in meerdere talen en maken deel uit van toonaangevende series, zoals *Best Practice*, *The Open Group series*, *Project management* en *PM series*.

Op de website van Van Haren Publishing is in de **Knowledge Base** een groot aanbod te vinden van whitepapers, templates, gratis e-books, docentmateriaal etc. Ga naar www.vanharen.net.

Van Haren Publishing is tevens de uitgever voor toonaangevende instellingen en bedrijven, onder andere: Agile Consortium, ASL BiSL Foundation, CA, Centre Henri Tudor, Gaming Works, IACCM, IAOP, IPMA-NL, ITSq, NAF, Ngi, PMI-NL, PON, The Open Group, The SOX Institute.

Onderwerpen per domein zijn:

IT en IT-management

ABC of ICT™
ASL®
CATS CM®
CMMI®
COBIT®
e-CF
ISO 17799
ISO 20000
ISO 27001/27002
ISPL
IT-CMF™
IT Service CMM
ITIL®
MOF
MSF
SABSA

Architecture (Enterprise en IT)

ArchiMate®
GEA®
Novius Architectuur Methode
TOGAF®

Business Management

BABOK® Guide
BiSL®
BRMBOK™
EFQM
eSCM
IACCM
ISA-95
ISO 9000/9001
Novius B&IP
OPBOK
SAP
SixSigma
SOX
SqEME®

Project-, Programma- en Risicomanagement

A4-Projectmanagement
DSDM/Atern
ICB / NCB
ISO 21500
MINCE®
M_o_R®
MSP®
P3O®
PMBOK® Guide
PRINCE2®

Voor een compleet overzicht van alle uitgaven, ga naar onze website: www.vanharen.net

Scrum Wegwijzer

Een kompas voor
de bewuste reiziger

Gunther Verheyen



Colofon

Titel:	Scrum Wegwijzer
Ondertitel:	Een kompas voor de bewuste reiziger
Auteur:	Gunther Verheyen
Reviewers:	Ken Schwaber (Scrum co-creator, Scrum.org) David Starr (Agile Craftsman, Microsoft) Ralph Jocham (Agile Professional, effective agile) Patricia Kong (Director of Program Development, Scrum.org) Paul Kuijten (Agile Coach, Agile Matters)
Tekstredactie:	Josien Moerman, Vertaal- en redactie bureau Am Rhein, Zwolle
Publisher:	Van Haren Publishing, Zaltbommel, www.vanharen.net
ISBN Hard copy:	978 94 018 0040 2
ISBN eBook:	978 90 018 0613 8
Druk:	Eerste druk, eerste oplage, april 2016
Layout en DTP:	CO2 Premedia, Amersfoort - NL
Copyright:	© Van Haren Publishing, 2014, 2016

Voor verdere informatie over Van Haren Publishing, e-mail naar: info@vanharen.net
Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, of op welke wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by the publisher.

Hoewel deze uitgave met veel zorg is samengesteld, aanvaarden auteur(s) noch uitgever enige aansprakelijkheid voor schade ontstaan door eventuele fouten en/of onvolkomenheden in deze uitgave.

Voorwoord door Ken Schwaber

An outstanding accomplishment that simmers with intelligence.

Scrum – A Pocket Guide is an extraordinarily competent book. Gunther has described everything about Scrum in well-formed, clearly written descriptions that flow with insight, understanding, and perception. Yet, you are never struck by these attributes. You simply benefit from them, later thinking, “That was really, really helpful. I found what I needed to know, readily understood what I wanted, and wasn’t bothered by irrelevancies.”

I have struggled to write this foreword. I feel the foreword should be as well-written as the book it describes. In this case, that is hard. Read Gunther’s book. Read it in part, or read it in whole. You will be satisfied.

Scrum is simple, but complete and competent in addressing complex problems. Gunther’s pocket guide is complete and competent in addressing understanding a simple framework for addressing complex problems, Scrum.

Een bewonderenswaardig werk bruisend van inzicht en kennis.

Deze *Scrum Wegwijzer* is een boek van uitzonderlijke kwaliteit. Gunther beschrijft alle aspecten van Scrum in heldere, duidelijke formuleringen. Zijn inzicht en expertise zijn evident, maar hinderen de eenvoud en duidelijkheid van de beschrijvingen niet. Je geniet, en leert, van zijn kennis en ervaring. Je komt meermaals op een punt dat je denkt: “Dat was echt, echt nuttig. Ik vond wat ik zocht, begreep het, en ik werd niet afgeleid door onnodige details of uitweidingen.” Het was een hele opgave om dit voorwoord te schrijven. Ik vond dat het voorwoord van dezelfde hoge kwaliteit moest zijn als het boek zelf. En dat bleek niet eenvoudig te realiseren. Lees het boek dat Gunther schreef. Lees het in gedeelten of in zijn geheel. Je zult aangenaam verrast zijn.

Scrum is eenvoudig maar volledig, en is geschikt om complexe problemen aan te pakken. Deze wegwijzer van Gunther is perfect om Scrum te begrijpen, een eenvoudig framework om complexe problemen aan te pakken.

Ken Schwaber, augustus 2013

Inleiding

De toepassing van Agile methoden voor softwareontwikkeling kent nog altijd een indrukwekkende groei. Scrum is de meest toegepaste methode voor Agile softwareontwikkeling. De interesse in Scrum is dan ook enorm.

Het is een hele uitdaging om de softwareontwikkeling van een organisatie met Scrum te gaan uitvoeren. Scrum is geen ‘proces’ in de klassieke zin van het woord, met gedetailleerde, alomvattende voorschriften voor elke denkbare situatie. Scrum schept een kader waarbinnen gewerkt kan worden, een *framework* gebaseerd op waarden en principes, en met rollen en regels gericht op de *mensen* die Scrum toepassen. Scrum komt pas tot leven als mensen het beginnen toe te passen. Scrum is een erg krachtig instrument, omdat het mensen in staat stelt om voortdurend opduikende nieuwe ideeën, mogelijkheden en technieken op te nemen. Scrum komt zelfs pas echt tot bloei als het pad van waanzinnig grootse, gedetailleerde plannen en voorspellingen wordt verlaten voor het pad van wat we in het Engels zo mooi omschrijven als *emergence*. Scrum gaat uiteindelijk meer over gedrag dan over proces.

Welke voordelen een organisatie creëert met Scrum hangt af van de wil en de vastberadenheid om oude regels en beperkingen op te heffen, om over grenzen heen te gaan denken en in te schepen op een ontdekkingsreis, een reis die leidt naar kansen en mogelijkheden.

Die reis start noodzakelijkerwijs met de toepassing van de regels van Scrum, wat kennis en inzicht in die regels vereist. Dit boek beschrijft deze regels. Het toont hoe Scrum concrete invulling geeft aan de Agile mindset, wat de fundamentele spelregels van Scrum zijn en hoe die regels ruimte laten voor een uitgebreid scala aan tactieken om het spel te spelen, om invulling te geven aan de regels. Het boek is een essentiële inleiding in alle mogelijke aspecten van Scrum, spelregels en speltactieken, de koppeling met ‘Agile’. Het legt daarbij sterk de nadruk op het begrijpen van de regels, het *waarom* ervan.

Dat is nuttig voor iedereen die Scrum nu reeds toepast, maar ook voor degenen die nog moeten beginnen aan de fascinerende reis die Scrum is.

Zelf ging ik in 2003 de weg op van agility via Scrum, een pad met vele hindernissen. Op mijn weg heb ik met vele teams mogen werken, in diverse projecten en voor sterk uiteenlopende klanten en gebruikers. Ik heb met kleine en grote organisaties mogen samenwerken en diverse onderdelen van organisaties mogen coachen: mensen, teams, coaches, managers. Het leverde me vele waardevolle contacten en relaties op, niet in het minst die met Ken Schwaber.

Ik dank die talloze mensen en teams voor de positieve energie.

Ik dank Ken Schwaber, David Starr, Ralph Jocham en Patricia Kong omdat ze vroege versies van dit boek mede vormgaven via constructieve, maar kritische reviews.

Ik dank het team van Van Haren Publishing voor het vertrouwen dat ze me schonken om dit boek te creëren.

Veel plezier met dit boek, en (vooral)... blijf Scrummen.

Gunther, juni 2013

Bij de Nederlandse vertaling

Dit werk werd in november 2013 gepubliceerd als Engelstalige uitgave getiteld *Scrum – A Pocket Guide (A Smart Travel Companion)*.

In de jaren voorafgaand aan deze uitgave heb ik het voorrecht genoten om in België – en sinds 2010 hoofdzakelijk in Nederland – kennis en inzicht in het Scrum-proces te mogen verspreiden. Vanuit mijn werk bij Scrum.org (sinds juni 2013) kreeg ik een goed beeld van de mondiale verspreiding van Scrum. Het doet me plezier vast te stellen dat Nederland wereldwijd (!) koploper is wat betreft de invoering van Scrum. Dankzij de vele mensen, teams en organisaties waarmee ik sinds 2010 mocht werken, is Nederland wat Scrum betreft zowat mijn thuisland geworden.

Vraag me niet waarom het zo lang geduurd heeft om mijn boek in mijn moedertaal te vertalen. Ik moet je het antwoord schuldig blijven. Ik hoop wel dat je mijn vreugde deelt dat het er eindelijk is, en vooral dat het je helpt om Scrum nog beter te begrijpen en onder de knie te krijgen. Moge het je helpen om betere software te ontwikkelen, en met meer energie en werkplezier je dagelijkse uitdagingen aan te gaan.

Ik dank Paul Kuijten voor het nalezen van mijn Nederlandse vertaling. Paul is een zielsverwant, een rebels agilist, een piraat met een peperkoeken hart. Zijn suggesties hebben er mede voor gezorgd dat mijn Scrum Wegwijzer zijn noodzakelijke eenvoud behield.

Bij de vertaling naar het Nederlands zijn kerntermen uit Scrum of gerelateerde vaktaal niet vertaald. Het moeilijk definieerbare begrip 'accountability' heb ik vertaald als 'aansprakelijkheid' en 'collaboration' als 'samen-werking'. Bij de vertaling van 'self-organization' heb ik gekozen voor 'zelf-sturing'. Ik heb in beide gevallen bewust voor het koppelteken gekozen, zodat de lezer duidelijk de combinatie van beide woorden proeft.

Blijf genieten. Blijf Scrummen.

Gunther Verheyen, februari 2016

Reviews

This Scrum Pocket Guide is outstanding. It is well organized, well written, and the content is excellent. This should be the de facto standard handout for all looking for a complete, yet clear overview of Scrum.

(Ken Schwaber, Scrum co-creator, Scrum.org)

Gunther has expertly packaged the right no-nonsense guidance for teams seeking agility, without a drop of hyperbole. This is the book about agility with Scrum I wish I had written.

(David Starr, Agile Craftsman, Microsoft)

During my many Scrum training activities I often get asked: “For Scrum, what is the one book to read?” In the past the answer wasn’t straight forward, but now it is! The Scrum Pocket Guide is the one book to read when starting with Scrum. It is a concise, yet complete and passionate reference about Scrum.

(Ralph Jocham, Agile Professional, effective agile)

‘The house of Scrum is a warm house. It’s a house where people are WELCOME.’ Gunther’s passion for Scrum and its players is evident in his work and in each chapter of this book. He explains the Agile paradigm, lays out the Scrum framework and then discusses the ‘future state of Scrum.’ Intimately, in about 100 pages.

(Patricia M. Kong, Director of Partners, Scrum.org)

Nederlandse vertaling

Deze Scrum Wegwijzer is een absolute hoogvlieger. Het is goed opgezet, knap geschreven en de inhoud is wonderbaarlijk. Dit is momenteel het standaardwerk met het meest complete, en toch beknopte overzicht van Scrum.

(Ken Schwaber, Scrum co-creator, Scrum.org)

Gunther heeft zijn expertise aangewend om teams richting te geven in hun zoektocht naar agility, rechttoe rechtaan, zonder overdaad of onnodig hoogdravende teksten. Dit is het boek over agility met Scrum dat ik graag zelf had geschreven.

(David Starr, Agile Craftsman)

Bij mijn Scrum opleidingen krijg ik vaak de vraag: “Wat is dat ene boek over Scrum dat ik absoluut zou moeten lezen?” Tot op heden was er geen eenduidig antwoord op die vraag. Nu wel! Deze Scrum Wegwijzer is dat ene boek dat je moet lezen. Het is een beknopte, maar complete beschrijving waar de passie van af straalt.

(Ralph Jocham, Agile Professional, effective agile)

‘Het huis van Scrum is een open huis. Het is een thuis waar mensen WELKOM zijn.’ De passie van Gunther voor het onderwerp Scrum blijkt duidelijk uit elk hoofdstuk, elke paragraaf van dit boek. Hij licht het Agile paradigma toe, geeft een complete schets van het Scrum framework en beschrijft de toekomst van Scrum. Krachtig, in slechts honderd bladzijden.

(Patricia M. Kong, Director of Program Development, Scrum.org)

Over deze Nederlandstalige versie

Dit boekje laat je VOELLEN wat zou kunnen met Scrum. Niet wat moet, maar wat zou kunnen! Sterke focus op het waarom. Waarom agile, waarom Scrum, waarom de rollen, waarom de events, waarom de artefacten. Zonder dat kun je het spel niet goed spelen, en dit boek beschrijft het waarom beter dan wat ik ooit gezien heb.

(Paul Kuijten, Agile Coach, Agile Matters)

Inhoud

Voorwoord door Ken Schwaber	V
Inleiding	VII
Reviews	XI
1 Het Agile paradigma.....	1
1.1 Een ingrijpende verschuiving	1
1.2 De oorsprong van Agile	6
1.3 Een definitie van Agile	7
1.4 Het iteratief-incrementeel continuüm	11
1.5 Agility kan niet gepland worden	14
2 Scrum.....	17
2.1 Scrum is een open huis.....	17
2.2 Scrum, of de betekenis van een naam	18
2.3 Wat zie ik daar, een gorilla?	19
2.4 Een framework, geen methodologie.....	22
2.5 Het spel spelen.....	26
2.6 Kernprincipes van Scrum	42
2.7 De kernwaarden van Scrum.....	52
3 Technieken versus regels	59
3.1 Visualisatie van voortgang.....	60
3.2 De vragen van de Daily Scrum.....	61
3.3 Product Backlog refinement	63
3.4 User Stories.....	64
3.5 Planning Poker	65
3.6 Sprint-lengte	66
3.7 Scrum op grotere schaal toegepast.....	68
4 De toekomst van Scrum	75
4.1 Ja, dit is Scrum. En	75
4.2 De mogelijkheden van het optionele product.....	77
4.3 Scrum stroomopwaarts	80

Bijlage A: Scrum begrippenlijst.....	85
Bijlage B: Literatuur.....	89
Trefwoordenregister	91
De auteur	95

Het Agile paradigma

1.1 Een ingrijpende verschuiving

Softwareontwikkeling werd heel lang gedomineerd door een wereldbeeld dat doordrongen is van industriële uitgangspunten en overtuigingen, een letterlijke vertaling van oude productietechnieken en theorieën naar een softwareomgeving. Aan de basis hiervan ligt een Taylor¹-achtig wantrouwen in uitvoerende arbeidskrachten, een wantrouwen dat impliceert dat het dergelijke ‘werkers’ aan voldoende intelligentie en creativiteit ontbreekt. Zij worden geacht enkel uitvoerend werk... uit te voeren. Daarom moeten hoger geplaatste personen – superieuren – de uitvoering voorbereiden, beschrijven en plannen. Ondanks de zorgvuldige voorbereiding moet er, volgens dit industriële model, dan nog steeds streng toegezien worden op de uitvoering van het werk. Kwaliteit wordt dan weer verzekerd door externe controle die plaatsvindt na de productiefase. Als dege-lijk beoordeeld werk wordt goedgekeurd en slechte partijen worden afgekeurd. Financiële beloningen en premies dienen om resultaten en gedrag in de gewenste richting te sturen. Op ongewenst gedrag volgen strenge straffen.

De meer dan ernstige tekortkomingen van deze aanpak in softwareontwikkeling zijn sinds lang bekend en beschreven. Met name in de Chaos-rapporten van de Standish Group is bij herhaling aangetoond dat deze aanpak nauwelijks succesvol is (Standish, 2015). De aanpak

¹ Frederick Taylor (1856-1915) was een Amerikaans ingenieur die bekendheid verwierf met zijn onderzoek en aanpak om productiviteit, efficiëntie en arbeidskosten te optimaliseren. Hij was een uitgesproken voorstander van opgelegde standaardisering en streng systematische methoden. In zijn visie lag alle controle volledig bij managers en was de rol van arbeiders strikt beperkt tot het uitvoeren van opgelegde opdrachten (bron: http://en.wikipedia.org/wiki/Frederick_Winslow_Taylor).



Figuur 1.1 Het industriële paradigma

van softwareontwikkeling volgens een industriële visie levert stapels problemen, incidenten en fouten op, die reeds lang de grenzen van het toelaatbare hebben overschreden. Maar jammer genoeg bestond de gangbare reactie op de vaststelling van deze problemen uit een bijsturing van de verwachtingen... naar beneden. Geleidelijk aan raakte het geaccepteerd dat slechts 10 tot 20% van de softwareprojecten succesvol is. ‘Succes’ wordt daarbij doorgaans gedefinieerd als: opgeleverd binnen het geplande tijdsbestek en budget, en voorzien van alle verwachte, vooraf gespecificeerde functionaliteit. *We kunnen vraagtekens plaatsen bij deze criteria van ‘succes’, maar het is wel wat het beschreven industriële denken als resultaat vooropstelt. Het is redelijk om de aanpak hierop af te rekenen.* Iedereen lijkt zich erbij neer te leggen dat de kwaliteit van softwareproducten laag is en dat meer dan 50% van de opgeleverde functionaliteit niet of nauwelijks gebruikt wordt (Standish, 2002).

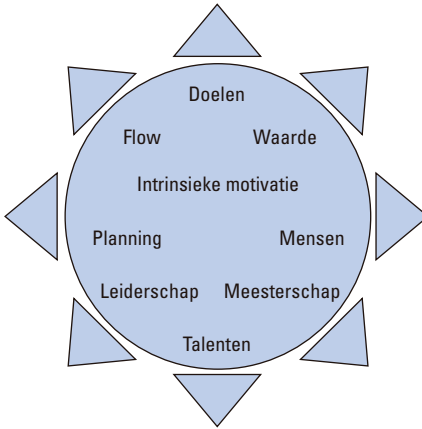
Het ligt gevoelig, maar het industriële paradigma heeft de software-industrie onmiskenbaar in een diepe crisis ondergedompeld. Velen hebben getracht uit het dal te klimmen door de industriële aanpak nog verder te versterken. Plannen werden nog gedetailleerder, meer fasen en overdrachten werden ingebouwd, nog meer tijd werd aan ‘ontwerpen vooraf’ besteed, evenals aan gedetailleerde specificaties en vele andere voorbereidende activiteiten. Dit alles gebeurde in de illusie dat het uiteindelijk uit te voeren werk beter gedaan kan worden als het vooraf compleet, gedetailleerd en met hoge precisie in beeld is

gebracht. De mate van detaillering vooraf werd immens opgevoerd. Steeds overheerste het idee dat de uiteindelijke uitvoerders nog meer en gedetailleerder aangestuurd moesten worden. Leidinggevendenden voerden het externe toezicht en de hiërarchische controle nog op. Het wantrouwen tegenover de ‘werker’ nam hand over hand toe. Elke stap, elke minuut werk, elke mogelijkheid werd dichtgetimmerd in documenten, verwachtingen en contracten.

Toch was er weinig verbetering merkbaar. De problemen bleven, het aantal defecten bleef hoog, de kwaliteit laag, en overschrijdingen van budget en doorlooptijd waren schering en inslag. Het beeld van softwareontwikkeling als een onbetrouwbare activiteit werd versterkt.

Het nam uiteindelijk heel wat tijd in beslag, maar onvermijdelijk doken nieuwe ideeën en inzichten op, vaak rechtstreeks gericht op de evidente problemen die het industriële paradigma veroorzaakte. De basis van een nieuw wereldbeeld werd uiteindelijk in de jaren '90 gelegd. In 2001 resulteerde dit in de term ‘Agile’ en in de formele beschrijving van een fundamenteel nieuwe visie op softwareontwikkeling. Het is een keerpunt in de geschiedenis. Het Agile denken voert tot de kern van het probleem en vervangt het industriële denken door een geloof in creativiteit en door respect voor de betrokken mensen en hun intelligentie. Het Agile denken erkent softwareontwikkeling als een creatieve en niet-repetitieve activiteit, die wordt uitgevoerd en beleefd door intelligente, zelfredzame kenniswerkers.

De software-industrie heeft meer dan voldoende redenen om het nieuwe denken met open armen te ontvangen. Enerzijds is er de omvang en wijde verspreiding van de beschreven problemen, anderzijds de context van het exponentieel toenemende belang van software in onze maatschappij. Het industriële denken in softwareontwikkeling blijkt sterk ingebakken te zijn. Ondanks de indrukwekkend slechte resultaten wordt het vandaag de dag zelfs nog steeds geprezen en onderwezen als dé aangewezen aanpak. Het aanvaarden van, en vervolgens handelen vanuit, een nieuw paradigma neemt echter ook gewoon tijd in beslag, veel tijd.



Figuur 1.2 Het Agile paradigma

Velen menen dat een radicale overgang naar Agile onmogelijk is en stellen een geleidelijke introductie van Agile technieken voor, binnen de klassieke aanpak. Toch is er voldoende reden voor scepsis met betrekking tot een dergelijke, geleidelijke overgang van het oude denken naar Agile.

Diepgaande verandering blijft doorgaans uit; er worden slechts oppervlakkige aanpassingen doorgevoerd, geen essentiële transformatie. Nieuwe termen worden ingevoerd, vaak zelfs opgelegd vanuit de hiërarchie, maar de fundamentele, onderliggende overtuigingen blijven onaangeroerd. Het eigenlijke gedrag verandert niet, het wordt slechts gemaskeerd met nieuwe namen en schijnwijzigingen. Essentiële tekortkomingen blijven daarmee onopgelost; niet in het minst het gebrek aan respect voor mensen en de voortdurende behandeling van creatieve, intelligente mensen als hersenloze uitvoerders.

Het behoud van de traditionele opvattingen heeft ook tot gevolg dat de nieuwe werkwijzen worden afgemeten aan oude normen. Echter, de concepten en ideeën van een fundamenteel nieuwe visie sluiten de voorgaande aanpak vaak compleet uit, en daarmee elk zinvol gebruik van bestaande meetmethoden en standaarden. Het vereist moed en ondernemerschap om de tekortkomingen van het oude wereldbeeld te onderkennen, en nog meer moed en gedurfd ondernemerschap om

deze te vervangen door een nieuwe aanpak, en het oude denken in al zijn aspecten definitief vaarwel te zeggen.

Pogingen om geleidelijk over te gaan van het industriële naar het Agile paradigma leiden tot een status quo, het behoud van het industriële paradigma.

Oorspronkelijk was veel van het bewijs dat Agile beter werkt, anekdotisch, gebaseerd op persoonlijke ervaringen, waarnemingen en meningen. In die zin was het Chaos Report dat de Standish Group in 2015 publiceerde, cruciaal. Voor het eerst werd op grote schaal vergelijkend onderzoek gedaan tussen op traditionele wijze aangepakte projecten enerzijds en projecten volgens de Agile benadering anderzijds. Het rapport gaf aan dat een Agile aanpak tot veel betere resultaten leidt, zelfs bij toetsing aan de oude definitie van succes als de combinatie van budget, tijd en scope. Het rapport toonde aan dat Agile projecten drie maal zo succesvol zijn als projecten volgens een traditioneel watervalproces. Bovendien bleek het aantal projecten dat dramatisch faalde slechts een derde te bedragen. Het mag wel duidelijk zijn dat toetsing aan meer hedendaagse succescriteria, zoals waardecreatie, allicht tot nog grotere verschillen zou leiden, met andere woorden dat het nieuwe, creatieve paradigma nog succesvoller zou blijken te zijn.

En toch, Agile is en blijft een keuze, geen verplichting. Het staat elke organisatie vrij deze weg wel of niet in te slaan. Het is één manier om de software-industrie tot een aangenamere plek te maken om dagelijks in te vertoeven. Onderzoek toont ook aan dat het tot meer succes leidt.

Scrum helpt.

De concrete regels van Scrum helpen om het nieuwe, Agile denken onder de knie te krijgen. De beperkte voorschriften en verplichtingen van Scrum, zoals beschreven in het volgende hoofdstuk, openen de mogelijkheid van directe actie, snelle stappen en resultaten. Dit versnelt de overgang naar het nieuwe paradigma aanzienlijk. Voor

een organisatie is Scrum een tastbare manier om zich het Agile paradigma eigen te maken. De weg hierheen bestaat uit voortdurende ontdekkingen, samenwerking en gecontroleerde leerprocessen, binnen het kader dat Scrum creëert. De organisatie en haar mensen ontdekken een nieuwe identiteit en gaan een toestand van *agility* in: een toestand die wordt gekenmerkt door continue verandering, wendbaarheid, verbetering en aanpassing. Scrum is een middel, geen doel.

Ondanks de eenvoud van Scrum toont de praktijk aan dat de invoering van Scrum vaak een erg grote verandering inhoudt. Misschien is dit te wijten aan de moeilijkheid om de oude zekerheden (alhoewel aantoonbaar schijnzekerheden) achterwege te laten. Misschien ligt het aan de tijd die het kost om een nieuw paradigma diepgaand te absorberen. Misschien ligt de oorzaak in de vastberadenheid en de volharding die het nu eenmaal vereist.

1.2 De oorsprong van Agile

Ondanks de overheersing van de plan-gedreven, industriële visie is een evolutieve aanpak van softwareontwikkeling niet nieuw. Craig Larman heeft de historische voorgangers van Agile uitgebreid beschreven in zijn boek *Agile & Iterative Development, A Manager's Guide* (Larman, 2004).

Maar de officiële benaming 'Agile' dateert van februari 2001. In deze periode werden de falende watervalprocessen vervangen door procedurele uitvoeringen van het theoretisch gezien iteratieve RUP-proces (*Rational Unified Process*), zonder substantiële verbeteringen. Zeventien leidende figuren uit de software-industrie kwamen samen in een skioord in Utah om de gemeenschappelijke elementen van hun visies te bespreken: Scrum, eXtreme Programming, Adaptive Software Development, Crystal, Feature Driven Development, DSDM, enzovoort. Aan de basis bleken ze heel wat fundamentele principes gemeen te hebben, en deze gaven ze het label 'Agile', na publicatie in het *Manifesto for Agile Software Development* (Beck et al., 2001).

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Veel teams, managers en organisaties spreken de wens uit om ‘Agile’ te worden. Al te vaak blijkt er onderliggend een verlangen uit naar een magische oplossing waarmee alle problemen als sneeuw voor de zon verdwijnen. Vaak zeg ik dan dat ‘Agile op zich eigenlijk niet bestaat’. Daarmee bedoel ik dat Agile niet één vastomlijnd proces is, één specifieke methode of aanpak. Agile is een verzameling van samenhangende principes die de verschillende methoden voor Agile softwareontwikkeling met elkaar gemeen hebben. ‘Agile’ verwijst naar de mindset, de overtuigingen en onderliggende principes beschreven in en geuit via het aangehaalde manifest.

Het Manifesto is een tastbare weergave van de waarden die aan Agile gedrag ten grondslag liggen. Iedereen die het Manifesto wil gebruiken om Agile te begrijpen, raad ik aan om voor een heel concrete invulling vooral ook naar de opgenomen twaalf principes te kijken.²

1.3 Een definitie van Agile

Zelf beschrijf ik ‘Agile’ graag aan de hand van de volgende sleutelkenmerken:

➔ Mensgericht

De sturing van Agile softwareontwikkeling gebeurt niet aan de hand van gedetailleerde plannen die aangeven hoe specificaties, die

2 Zie: <http://agilemanifesto.org/principles.html>

volledig en op voorhand ontleed, ontworpen en beschreven zijn, geïmplementeerd moeten worden. Agile onderkent de onmogelijkheid om software-requirements compleet op voorhand te kunnen voorspellen, en daarmee dus ook de onmogelijkheid om voorafgaand aan de eigenlijke implementatie een perfect design, een complete architectuur en dergelijke op te maken.

Een Agile proces maakt korte metten met de vele overdrachten van tussentijdse documenten, beschrijvingen of akkoorden tussen niet-samenwerkende en geïsoleerde diensten en specialisaties. Agile wordt gedreven door de voortdurende *samenwerking* en interactie tussen mensen, ongeacht het departement, de afdeling of de dienst waar ze aan verbonden zijn. Evenmin maakt het uit of deze mensen deel uitmaken van business, IT, marketing, sales, operaties of management. De talenten en inzichten van de betrokken mensen zijn van belang, en de uitwisseling van hun ideeën en inzichten. Hun positie of hun uiterlijk doet niet ter zake, net zomin als de vraag wie hun salaris betaalt.

Mensen worden gewaardeerd en gerespecteerd omwille van hun creativiteit, probleemoplossend vermogen en kunde om samen te werken en hun werk zelfstandig te organiseren. Van belang zijn hun capaciteiten om een probleem te begrijpen en aan te pakken zonder overdadige sturing, controle en bureaucratie. Ceremoniële activiteiten (meetings, afspraken, overdrachten, handtekeningen) zijn geen gelijkwaardig equivalent van creatief denken, innovatie, aansprakelijkheid en verantwoordelijkheid. Sterker nog, meestal staan ze deze in de weg.

Een menselijke aanpak wordt gerespecteerd via het principe van draaglijke voortgang (*Sustainable Pace*). Mensen krijgen de kans hun werk zo te organiseren dat het tempo en de voortgang ervan draaglijk zijn, op korte en op lange termijn.

→ Dienend leiderschap

De fundamenteën van Agile bieden geen plaats aan de klassieke recepten van hiërarchische superioriteit, het toewijzen van uitvoerbare

taken aan slaafse, individuele medewerkers, micromanagement en totalitaire machtsuitoefening.

In Agile worden mensen en teams gefaciliteerd via dienend leiderschap (*servant leadership*). Binnen afgesproken grenzen en kaders regelen mensen en teams hun werk en taken zelf. Doelstellingen en visie geven de algemene richting aan, niet van tevoren bepaalde microtaken. Controle is subtiel, en ontstaat vanuit de kaders.

→ **Iteratief-incrementeel**

Agile processen zijn niet vrijblijvend. Ze zijn duidelijk gedefinieerd en vereisen veel discipline.

Producten worden in zinvolle onderdelen uitgevoerd en opgeleverd, waarbij elk nieuw onderdeel voortbouwt op het vorige, door de wijziging en verbeteringen van bestaande onderdelen en toevoeging van nieuwe. Elk onderdeel waaraan gewerkt wordt, wordt voortdurend bewaakt en desgewenst aangepast om de integriteit en goede samenhang ervan te garanderen. Datzelfde geldt voor het totaalproduct waartoe het behoort.

Agile legt expliciet veel nadruk op kwaliteit en hoogwaardige resultaten, met eens te meer het uitgangspunt dat kwaliteit niet louter in documenten en papieren teksten kan worden gegoten. Kwaliteit kan ook niet aan een afgewerkt product toegevoegd worden, maar moet in het gecreëerde product ingebouwd worden.

→ **Succes**

Voortgang kan in softwareontwikkeling niet vastgesteld of gegarandeerd worden aan de hand van vooraf bepaalde plannen en mijlpalen, documenten, overdrachten, ondertekeningen, meetings, of andere louter administratieve gebeurtenissen. Dat zijn overblijfselen van de industriële visie op softwareontwikkeling. Agile erkent werkende software die echt af is als de ultieme maat voor voortgang, en de mate waarin het product een verschil maakt voor de eindgebruikers.

Pas op het moment dat ze het (deel)product effectief kunnen gebruiken, kunnen gebruikers volledig zekerheid geven over hun

verwachtingen en hun tevredenheid. Dat is eigen aan software. Papierwerk kan nooit werkelijk gebruik van een product vervangen.

Agile erkent geen scheiding tussen business en IT: beide zijn noodzakelijk voor succes. Zonder actieve aanwezigheid van beide perspectieven kan software niet én bruikbaar én nuttig zijn.

→ **Verandering**

Zelfs al zou men specificaties compleet en gedetailleerd op voorhand kunnen vastleggen, dan nog blijven ze aan verandering onderhevig. Markten en bedrijfsstrategieën veranderen, de concurrentie staat niet stil, gebruikers begrijpen pas echt de mogelijkheden van software als ze er effectief mee aan de slag kunnen. En dat zijn slechts enkele van de veranderingen die kunnen optreden en de voortgang van een softwaretraject beïnvloeden.

In een voorspellende, planmatige aanpak wordt verandering als een bedreiging afgeschilderd, als een negatief element. Vanwege de bedreiging die ze voor in beton gegoten plannen vormen, worden veranderende meningen of inzichten ontkend of verbannen naar externe comités en vergaderingen. Ze worden buiten het eigenlijke werkproces geduwd.

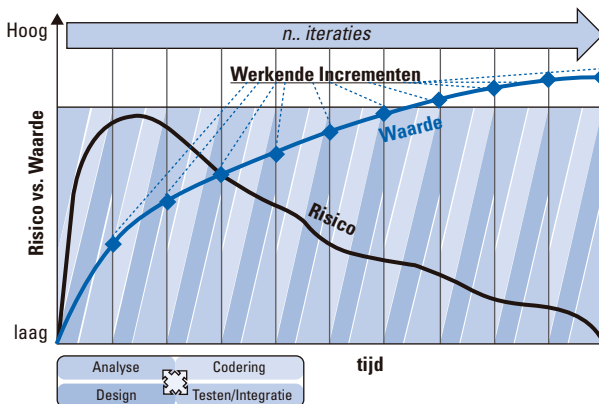
De Agile werkwijze beschouwt verandering echter niet alleen als normaal, maar zelfs als verrijkend. Nieuwe inzichten, veranderende meningen en wijzigingen zijn cruciaal. Agile wordt voortgedreven door 'emergence': het onverwacht opduiken van voorheen onbekende verwachtingen, plannen, ideeën, inzichten, architecturen, designs. Verandering werkt niet verstorend, omdat het een ingebakken onderdeel van een Agile proces vormt. In een Agile werkomgeving worden veranderingen zelfs aangemoedigd als een bron van innovatie en verbetering.

Het negatieve karakter van 'change' verdampt, 'change' bestaat niet meer, lost als het ware op in het niets.

1.4 Het iteratief-incrementeel continuüm

Elk Agile proces deelt de tijd op in ‘time-boxed’ iteraties: perioden met een vast start- en eindpunt. Deze vorm van timemanagement biedt talloze voordelen, waarbij *focus* toch wel een van de belangrijkste is. Elke iteratie heeft als doel om een deelproduct te maken dat uit een werkende productversie bestaat, zodat de werking ervan kan worden geëvalueerd en er op basis van die toetsing geleerd en bijgestuurd kan worden. Met time-boxing wordt de zekerheid ingebouwd dat het verrichte werk op gezette tijden wordt geïnspecteerd. Daardoor kunnen nog tijdens de lopende werkzaamheden de nodige lessen getrokken én verwerkt worden, in de volgende iteratie.

Agile softwareontwikkeling wordt aangestuurd door de *business*, en via de business door kansen die onder meer de markt biedt. In een traditionele aanpak organiseert men de werkzaamheden die inherent aan softwareontwikkeling zijn (analyse, design, ontwikkeling, release, test, documentatie) sequentieel en in langlopende, aansluitende fasen. In Agile is de organisatie van deze zelfde werkzaamheden echter compleet anders. Ze worden in parallel uitgevoerd, en in elke iteratie opnieuw, zodat men snel en flexibel op nieuwe en veranderende marktverwachtingen en –kansen kan inspelen.



Figuur 1.3 Waardecreatie in Agile