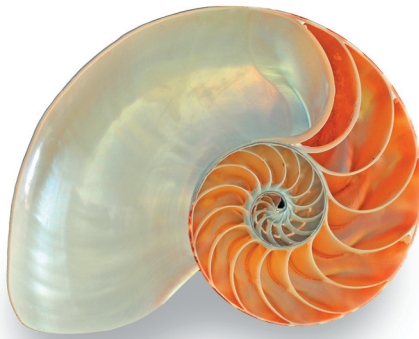


HENNIE HUIJGENS

Agile werkt



*Meten, analyseren
en benchmarken
van IT-projecten*

Agile werkt

Meten, analyseren en benchmarken van IT-projecten

Hennie Huijgens



Meer informatie over deze en andere uitgaven kunt u verkrijgen bij:
Sdu Klantenservice
Postbus 20014
2500 EA Den Haag
tel.: (070) 378 98 80
www.sdu.nl/service

© 2012 H. Huijgens

Academic Service is een imprint van Sdu Uitgevers bv.

Zetwerk: Fritschy opmaak & redactie, Leiden
Omslagontwerp: VillaY, Den Haag

ISBN 978 90 12 58127 1
NUR 980

Alle rechten voorbehouden. Alle intellectuele eigendomsrechten, zoals auteurs- en databankrechten, ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij Sdu Uitgevers bv en de auteur.

Behoudens de in of krachtens de Auteurswet 1912 gestelde uitzonderingen, mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 3051, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk)fouten en onvolledigheden niet worden ingestaan en aanvaarden de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the publisher's prior consent.

While every effort has been made to ensure the reliability of the information presented in this publication, Sdu Uitgevers neither guarantees the accuracy of the data contained herein nor accepts responsibility for errors or omissions or their consequences.

Inhoud

Inleiding 13

Deel 1 *Agile* projecten leveren betere resultaten op dan traditionele

1 Het draait allemaal om planning 21

- 1.1 Meten en analyseren omwille van procesverbetering 21
- 1.2 Een verrassende wending door een focus op *Scrum* 22
- 1.3 Geplande verbetering of een toevalstreffer? 23
- 1.4 De periode voorafgaand aan *Scrum* 24
 - 1.4.1 Dilemma 1: experts plannen geen verbetering 25
 - 1.4.2 Dilemma 2: het managen van onzekerheden staat niet op de agenda 26
 - 1.4.3 Dilemma 3: managers sturen op uitnutting van kosten 28
- 1.5 Het resultaat: planning gerealiseerd, performance verslechterd 28
- 1.6 Maar er gloort hoop aan de horizon: *agile* werkt! 29
- 1.7 Samenvatting 30

2 De wereld van de beslisser: ratio en non-ratio 31

- 2.1 Indicatoren voor IT-performance 33
 - 2.1.1 Time-to-market 34
 - 2.1.2 Productiviteit 34
 - 2.1.3 Kwaliteit 35
- 2.2 Een lerende organisatie 36
- 2.3 Een invulling van de MA-leercyclus volgens de basismetrieën 38
- 2.4 Een geïntegreerde benadering van MA: IT-portfoliomanagement 38
- 2.5 Nog een geïntegreerde benadering van MA: IT-risicomanagement 39
- 2.6 De wereld verandert – en de IT-wereld verandert mee 42
- 2.7 Samenvatting 44

3	De theorie achter kwantificeren van de IT-performance	45
3.1	Een standaardset van IT-metrieken	45
3.2	Five Core Metrics	46
3.2.1	Omvang (<i>size</i>)	48
3.2.2	Funcatiepuntanalyse	49
3.2.3	Omvang meten in de praktijk	50
3.2.4	Source Lines of Code (SLOC)	51
3.2.5	<i>Backfiring</i> ; omrekenen van functiepunten naar SLOC	52
3.2.6	Inspanning en doorlooptijd	52
3.2.7	Kwaliteit	53
3.3	Het kwantificeren van IT-portfolio's en projectplanningen	54
3.4	Omgaan met onzekerheid	56
3.5	Focus op procesverbetering	57
3.5.1	Capability Maturity Model Integrated (CMMI)	57
3.5.2	De uitdaging van CMMI: 'Wat ga je meten?'	59
3.5.3	Six Sigma	60
3.5.4	Six Sigma: verkleinen van de standaardafwijking	60
3.5.5	Lean Six Sigma	61
3.6	Kwantificeren van (nieuwe) delivery-modellen	61
3.6.1	RUP	63
3.6.2	DSDM Atern	64
3.6.3	<i>Scrum</i>	64
3.6.4	Het planningsproces bij <i>agile</i> ontwikkelmodellen	66
3.7	Meten van de marktconformiteit van IT-projecten	68
3.8	Samenvatting	70

Deel 2 Een onderzoek naar de performance van IT-projecten

4	De opzet van het onderzoek	75
4.1	Het onderwerp en de centrale vraag van het initiële onderzoek	75
4.2	Het onderwerp en de centrale vraag van het vervolgonderzoek	77
4.3	De gebruikte onderzoeksmethode	77
4.4	Het onderzoeksmodel	78
4.5	Benchmark uitgangspunten	79
4.6	De onderzochte softwareontwikkelingsorganisaties	80
4.7	De organisatie rond meten en analyseren	81

4.8	Agile development (<i>Scrum</i>)	82
4.9	Een kritische beschouwing van de gebruikte onderzoeksmethode	85
4.10	Samenvatting	86
5	Dataverzameling	89
5.1	Waarnemingen	89
5.1.1	Afgeronde projecten	90
5.1.2	Lopende projecten	90
5.2	Verdeling van projecten over softwareontwikkelingscategorieën	91
5.3	Validiteit en betrouwbaarheid	91
5.4	Een kritische beschouwing van de dataverzameling	92
5.5	Samenvatting	93
6	Analyse van de basismetrieën	95
6.1	Beschrijvende statistiek	95
6.2	Nadruk op het onderkennen van trends	96
6.3	Statistische onderbouwing van correlatie en standaardafwijking	96
6.3.1	Sigma	97
6.3.2	Trendlijnen	97
6.4	De basismetrieën: omvang, doorlooptijd, inspanning, kosten	98
6.4.1	Omvang	98
6.4.2	Doorlooptijd	101
6.4.3	Inspanning	104
6.4.4	Kosten	105
6.5	Analyse van kosten versus doorlooptijd: effecten van tijdcompressie	107
6.6	Fouten	109
6.7	Samenvatting van de marktconformiteit	111
7	Analyse van de performance-indicatoren	113
7.1	Productiviteit	113
7.1.1	Kosten per functiepunt	113
7.1.2	Kosten per functiepunt bij waterval en agile projecten	115
7.1.3	Uren per functiepunt	115
7.1.4	Productiviteitsindex (PI)	115
7.1.5	Manpower Buildup Index (MBI)	118

7.2	Time-to-market	118
7.2.1	Doorlooptijd per functiepunt bij waterval en agile projecten	119
7.3	Kwaliteit	120
7.4	De kengetallen voor de performance-indicatoren samengevat	121
7.5	Succesfactoren: karakteristieken van <i>best practices</i>	122
7.6	Faalfactoren: karakteristieken van slechte presteerders	124
7.7	Samenvatting en conclusies	125
8	Aanvullend onderzoek naar succesfactoren	127
8.1	Voorspelbaarheid	128
8.1.1	Voorspelbaarheid van kosten	128
8.1.2	Voorspelbaarheid van doorlooptijd	130
8.1.3	Voorspelbaarheid van kosten versus voorspelbaarheid van doorlooptijd	131
8.2	Planningsaspecten	131
8.2.1	De gehanteerde planningsmethode	132
8.2.2	Geplande productiviteit van lopende projecten	133
8.2.3	Geplande time-to-market van lopende projecten	134
8.2.4	Het planningsprobleem samengevat	134
8.3	Schaalgrootte	135
8.4	Het effect van het gekozen ontwikkelmodel	137
8.5	Samenvatting en conclusies	138

Deel 3 Hoe voer je een *agile* manier van meten, analyseren en benchmarken in?

9	Een basismodel voor meten, analyseren en benchmarken	141
9.1	Vijf fasen voor meten, analyseren en benchmarken	142
9.1.1	Fase 1; anarchie	142
9.1.2	Fase 2: meten van de performance van IT-projecten	143
9.1.3	Fase 3: organisatiebrede aanpak gericht op benchmarken van het IT-portfolio	143
9.1.4	Fase 4: businessgeoriënteerde aanpak van portfolio- en risicomanagement	143
9.1.5	Fase 5: innovatie	144

10	De MA-leercyclus vertaalt naar de praktijk	147
10.1	MA-Projectafsluiting	147
10.2	MA-Benchmark en -Rapportage	149
10.3	MA-Projectplanning	149
10.3.1	What-if scenario's	150
10.3.2	Trade-off tussen tijd, geld en kwaliteit	151
10.3.3	Tijdcompressie	151
10.3.4	Grootte van het ontwikkelteam	152
10.3.5	Schaalgrootte	152
10.3.6	Ontwikkelmethoden: agile versus waterval	153
10.3.7	Topdown versus bottom-up	153
10.3.8	Omgaan met onzekerheid	154
10.4	MA-Projectbeheersing	155
11	Een geïntegreerde aanpak voor meten en analyseren	159
11.1	Kwantificeren van IT-leveranciersmanagement	159
11.1.1	Een bonus/malusaanpak gebaseerd op de performance-indicatoren	160
11.1.2	Analyseren van leveranciersoffertes	161
11.2	Van een projectscope naar een portfolioaanpak	162
11.2.1	Denk als een varken	163
11.2.2	Brutoproductiviteit versus nettoproductiviteit	163
11.2.3	Optimaliseren van portfolio's door schaalgrootte	164
11.3	Kwantificeren van de business case	165
11.4	Een kwantitatieve aanpak van IT-risicomanagement	166
11.4.1	Een risicoaanpak op basis van projectmonitoring	167
11.4.2	Kwadrant A: geen bijsturing nodig	168
11.4.3	Kwadrant B: verzekeren van projectrisico's	169
11.4.4	Kwadrant C: maatregelen om projectrisico's te mitigeren	169
11.4.5	Kwadrant D: wanneer het echt fout gaat; uitwijkscenario's	169
12	Invoeren van een meet- en analysecompetentie	171
12.1	Een blauwdruk voor prioriteitsstelling	171
12.2	Een raamwerk voor metrieken	173
12.2.1	Metrieken voor het meten van output	175
12.2.2	Metrieken voor het meten van input	176

12.2.3	Procesmetrieën	177
12.2.4	Metrieën voor het meten van waardecreatie	179
12.3	Organisatieaspecten	180
12.3.1	Geautomatiseerde hulpmiddelen die meten en analyseren ondersteunen	180
12.3.2	Organisatie van de workload van een meet- en analyseteam	181
12.3.3	Organiseer je team op een agile manier	182
12.3.4	Rollen en verantwoordelijkheden in een MA-team	182
12.4	Maatregelen die de performance van IT een 'boost' geven	184
12.4.1	Een geïntegreerde visie op verbeteren en kwantificeren van doelen	185
12.4.2	Meten, analyseren en benchmarken verloopt gefaseerd	187
12.4.3	Zes generieke adviezen voor performanceverbetering	188
12.4.4	<i>Go agile!</i>	189

Dankwoord	191
------------------	------------

Noten	193
--------------	------------

Namenregister	201
----------------------	------------

Trefwoordenregister	203
----------------------------	------------

Inleiding

Dit is een positief verhaal. Een verhaal over een nieuwe en moderne manier om informatiesystemen te bouwen en te onderhouden. Het is opgebouwd uit drie delen: (1) *agile* projecten leveren betere resultaten op dan traditionele; (2) onderzoek wijst dat uit en ondersteunt dat, en (3) hoe voer je een *agile* manier van meten, analyseren en benchmarken van die projecten in? Maar ja, zoals zoveel goede verhalen begint het met ellende. Lees een krantenartikel of een column in een vakblad over informatietechnologie en de nadruk ligt in al die keren dat IT-projecten mislukten, op de projecten die veel duurder werden dan ze gepland waren of op de projecten waarbij het beloofde resultaat maar voor de helft en veel te laat werd opgeleverd. Capers Jones, de auteur van het eerste boek dat ik ooit las over het meten van IT-projecten, begint een artikel over *benchmarking* door zonder omhaal te stellen dat de softwarebranche geen goede reputatie heeft voor het op tijd leveren van oplossingen, het beheersen van de kosten en het leveren van kwaliteit.¹ En in een verslag van Chris Verhoef van de Vrije Universiteit in Amsterdam over een rondetafelgesprek over grote IT-projecten bij de overheid, lees je dat hij een toegenomen transparantie door rapportages ziet, maar tegelijkertijd merkt hij op dat we juist door die rapportages zien dat de IT-praktijk bij de overheid niet goed genoeg is.² Oké, ze hebben allebei gelijk, maar er zijn ook voorbeelden uit de praktijk die juist tonen hoe je wel die verbetering kunt realiseren. De praktijk bewijst dat het beter kan en in dit boek laat ik zien wat de kritieke succesfactoren voor die verbetering zijn en hoe je dat succes daarna ook kunt aantonen.

Dit boek kent twee verhaallijnen. Zoals de titel al doet suggereren gaat het over de kracht van *agile* werken. *Agile*, dit woord betekent in het Engels lenig of behendig, is een conceptueel raamwerk voor het in korte sprints en kleine, zelfsturende ontwikkelteams ontwikkelen van software. *Agile* is een alternatief voor meer traditionele ontwikkelmethoden die doorgaans met de term waterval of lineair worden aangeduid. Het verbeteren van de performance van IT-projecten door *agile* te gaan werken is misschien wel de belangrijkste boodschap van dit boek. De doelstelling is om organisaties die IT-oplossingen maken te helpen om dat goedkoper, beter en sneller te doen. En de boodschap is simpel: start morgen met *agile* werken en je zult merken dat je performance verbetert. Maar de ondertitel suggereert ook een tweede boodschap: dit boek gaat naast *agile* ook over meten, analyseren en benchmarken van IT-projecten. Want alleen dan wanneer je de resultaten van je werk meet en analyseert kun je ook daadwerkelijk leren en continu blijven verbeteren.

En alleen wanneer je meet en analyseert kun je aan alle betrokkenen echt laten zien dat je IT-performance is verbeterd. Dus de eerste verhaallijn laat aan de hand van praktijkvoorbeelden en de resultaten van wetenschappelijk onderzoek zien dat *agile* werken een uitstekende en voor de hand liggende keuze is. En de tweede verhaallijn gaat over hoe je in zo'n *agile* werkomgeving een meetprogramma invoert. Wat moet je doen en wat juist niet? Welke metriekeken zijn belangrijk en waarom? Wat zijn de valkuilen en wat zijn de succesvoorwaarden voor een goede en werk-bare aanpak voor meten en analyseren van IT-projecten? Het mag dus duidelijk zijn dat *agile* en meten en analyseren onlosmakelijk met elkaar zijn verbonden. En dat dit boek dus eigenlijk één echte boodschap heeft. Vandaar de titel: *Agile werkt*.

En er is nog een plezierige, absoluut niet onbelangrijke bijkomstigheid om *agile* te gaan werken. De mensen die het echte werk doen – de ontwerpers, de programmeurs, de testers – zullen weer blij en trots op hun werk zijn. Nu wil ik niet beweren dat mensen die aan watervalprojecten werken dat per definitie niet zijn. Maar het valt me regelmatig op dat juist degenen die in een *agile* omgeving werken nadrukkelijk uitspreken dat ze hun werk leuk vinden. Ze zijn vaak meer dan in een watervalomgeving direct betrokken bij de besluitvorming over wat er de komende twee weken gedaan wordt. Ze praten en overleggen dikwijls direct met de opdrachtgever. En misschien juist daardoor wordt er meer naar ze geluisterd. In watervalprojecten zie je toch vaak dat programmeurs en testers ver weg van de opdrachtgever aan het werk zijn. Ze hebben vaak uitsluitend contact met IT-managers en ze hebben weinig invloed op de besluitvorming. De besluiten worden genomen in vergaderingen waaraan zij meestal niet deelnemen. Bij *agile* is dat anders. Daar komt de opdrachtgever langs tijdens demo's aan het einde van elke sprint en kunnen de programmeurs laten zien wat zij hebben gedaan. De testers kunnen aangeven welke resultaten zijn bereikt en tegen welke hindernissen zij aanlopen. En aan het begin van een sprint bepalen de specialisten zelf hoeveel werk zij in de komende sprint gaan uitvoeren. Begrippen als creativiteit, zelfsturing en verantwoordelijkheid zijn niet vanzelfsprekend verbonden met waterval. Maar in een *agile* omgeving zijn ze bijna vanzelfsprekend. En daar worden vakmensen blij van. Een *Scrummaster*, iemand die een *agile* team begeleidt en die ervoor zorgt dat het bouwproces probleemloos verloopt, vertelde me dat hij 'sinds jaren weer echt plezier in zijn werk had'. En dat is geen onbelangrijk neveneffect van *agile*, gezien de berichten in de vakpers over groeiende tekorten aan gekwalificeerde vakmensen.³

Agile gaat over het doen van de juiste dingen. En over het niet doen van onnodige dingen. Anders gezegd, een doelstelling van *agile* is het zoveel mogelijk voorkomen van verspilling. Aan elke fout die een programmeur maakt moet een tester tijd besteden. En zo'n fout moet weer hersteld worden en dan nog een keer getest. Iets in één keer goed doen voorkomt verspilling; dat is de achterliggende

gedachte van *agile*. Soms maken mensen daarbij de denkfout dat allerlei ondersteunende activiteiten in een project daarom ook automatisch verspilling zijn. In het verlengde van die gedachte zou je kunnen denken dat in een *agile* omgeving ook het meten en analyseren van de performance van de ontwikkelteams niet nodig is. Dat het een overbodige luxe is die niet noodzakelijk is voor het bouwen van software en daarom als verspilling beschouwd kan worden. Wat mij betreft een grote misvatting. Wanneer je niets meet, dan is het succes van een ontwikkelteam ook niet aantoonbaar. Dat succes bestaat dan vaak eenvoudigweg niet voor veel van de betrokkenen. En succes is belangrijk, want door succes creëer je draagvlak en begrip bij de beslissers. Wanneer een organisatie besluit om *agile* te gaan werken, dan vraagt zo'n besluit om onderbouwing en bewijs. Dus meten, analyseren en benchmarken van IT-projecten is een *must*.

Al meer dan twintig jaar help ik grote informatie-intensieve organisaties – banken, verzekeraars, overheid en semioverheid, onderwijsinstellingen, zakelijke dienstverlening – bij het behalen van meerwaarde uit hun IT-projecten door te kijken naar de performance van afgeronde projecten en daaruit lessen te leren voor nieuwe activiteiten. Ik heb in mijn dagelijkse praktijk als *measurement* consultant vaak gezien welke problemen ontstaan wanneer *Measurement & Analysis* – zoals het proces waarin het meten en analyseren van projecten vaak wordt genoemd – wordt ingevoerd in een softwareontwikkelomgeving: te complex, te veel in één keer, of misschien wel de juiste informatie maar op een totaal verkeerd tijdstip. Kortom: niet *agile*.

Een voorbeeld. Een senior managementteam vroeg mij eens om een metriek in te voeren die iets moest zeggen over de betrouwbaarheid van de *high impact* projectplanningen, de projectplanningen die werden gemaakt in een zeer vroeg stadium van de IT-projecten, terwijl men eigenlijk geen gegevens verzamelde en bewaarde over de *business requirements* en over de voortgang van projecten in die voorfase. Een plan dat bij voorbaat gedoemd was om te mislukken. In plaats van zich bezig te houden met zo'n superspecifiek onderdeelje van de IT-projecten zouden de managers zich moeten afvragen wat hun overall performance nu eigenlijk was. Het lijkt zo voor de hand liggend, maar het stellen en beantwoorden van de juiste vraag op het juiste tijdstip is een specialisme. Voor je het weet holt iedereen in een organisatie achter een paar compleet verkeerde metrics aan en wordt het proces om die metrics te verzamelen en te analyseren tot een heilige graal verklaard. Het kan nog erger; huur een adviseur in die je gaat begeleiden in zo'n zoektocht, hij zal het graag een *queste* noemen, en je weet zeker dat de reis erg mooi is maar dat de kans groot is dat je de verkeerde dingen gaat verzamelen en meten.

Dit boek helpt meet- en analysespecialisten maar ook IT-managers en -opdrachtgevers om de invoering van een meet- en analyseprogramma stap voor stap te

plannen. Het laat zien hoe je daarbij zo goed als mogelijk kan aansluiten op zowel de behoefte aan informatie van de beslissers, als op de vrijheid van werken van ontwikkelteams. Het boek leert je welke acties en bijbehorende metrics een echte verbetering kunnen faciliteren, zonder dat de specialisten in de ontwikkelteams het gevoel krijgen dat zij allerlei overbodige en gedetailleerde informatie moeten verzamelen louter en alleen voor het plezier van een enkele *measurement nerd*.

Agile werkt is bedoeld voor een brede en diverse doelgroep. Allereerst is het gericht op al die professionals die dagelijks bezig zijn met het ontwikkelen en onderhouden van complexe IT-systemen en hun managers die verantwoordelijk zijn voor het creëren van de optimale omstandigheden daarvoor. Het boek is een *must read* voor CIO's en opdrachtgevers omdat het hen helpt met het adresseren van de acties die de verbeteringen in softwareontwikkeling moeten versnellen. Het laat ze zien hoe zij goede en bruikbare planning en beheersing van de IT-projecten in hun IT-organisaties kunnen faciliteren. En behalve dat kan het boek uitstekend worden gebruikt als een leerboek en handleiding voor meet- en analyseprofessionals en informatietechnologie- en vooral ook informatiemanagementstudenten aan zowel hogescholen, universiteiten als business schools.

De naam van dit boek simpelweg is *Agile werkt*. Je zou kunnen denken dat ik met die naam wel erg overduidelijk appelleer aan de relatieve hype die *agile* is. Oké, ik ben het ermee eens dat ik de reclameboodschap ervan absoluut op waarde schat. Maar de echte reden is dat ik oprecht geloof – en onderzoek onderbouwt de houdbaarheid van dit statement – dat *agile* echt werkt. Zoals ik zal aantonen in dit boek. Om alvast de toon te zetten wat snelle cijfers: onderzoek binnen twee representatieve organisaties van 296 IT-projecten laat zien dat *agile* projecten 34% sneller dezelfde hoeveelheid functionaliteit opleverden dan traditionele watervalprojecten, en dat de kosten van die projecten gemiddeld 27% lager waren. En de kwaliteit van het ontwikkelproces, gemeten in het aantal fouten, is bij *agile* projecten 21% beter dan in een watervalomgeving. Hoeveel redenen heb je nog meer nodig om niet morgen al te starten met *agile* projecten?

Het boek is georganiseerd in drie delen. Deel 1 is gebaseerd op de overtuiging dat een andere – meer *agile* – kijk op softwareontwikkeling in een organisatie vraagt om een andere – meer *agile* – aanpak van performancemanagement. Ik onderbouw die hypothese door een drietal dilemma's te schetsen dat bepalend is voor het falen van echte verbeteringen in een IT-organisatie en door te tonen waarom dat in een *agile* werkomgeving anders verloopt. Daarnaast bevat Deel 1 een hoofdstuk dat ingaat op de belangrijkste theorie met betrekking tot performancemanagement van informatietechnologie. Deel 1 bevat de basisfilosofie van het gehele boek, dus als je verder niets wilt lezen en toch een beeld wilt hebben van de beweegredenen achter dit boek, lees dan dit deel.

Deel 2 is opgebouwd rond een onderzoek dat ik heb uitgevoerd op basis van projectdata van bijna driehonderd representatieve IT-projecten van twee grote, informatie-intensieve organisaties. Dit onderzoek was feitelijk een vervolg op een wetenschappelijke studie die ik in het kader van mijn studie aan de Universiteit van Amsterdam in 2011 heb uitgevoerd binnen een grote, informatie-intensieve serviceorganisatie naar de performance van de softwareontwikkelingsactiviteiten. Het onderzoek werd uitgevoerd binnen een aantal qua aandachtsgebied verschillende softwareontwikkelingsafdelingen van een van de onderzochte organisaties, maar de meerderheid van de bevindingen van het onderzoek bevat generieke aspecten die waarschijnlijk ook van toepassing zijn op andere informatie-intensieve organisaties. Om te achterhalen in hoeverre de resultaten van dit onderzoek ook van toepassing zijn op andere bedrijven heb ik voor dit boek een nieuw, aanvullend onderzoek uitgevoerd. Voor dit aanvullende onderzoek heb ik gebruik gemaakt van een grote set van gegevens die verzameld zijn binnen een vergelijkbare informatie-verwerkende organisatie die grootschalige IT-projecten uitvoerde. Omwille van de objectiviteit en vertrouwelijkheid zijn zowel alle projectgegevens als de gegevens die gerelateerd zijn aan de onderzochte organisaties in dit boek geanonimiseerd. De relevantie van het oorspronkelijke onderzoek lag hoofdzakelijk in het feit dat binnen de initieel onderzochte organisatie zowel de IT-organisatie als de opdrachtgevende organisatie aangaf onvoldoende inzicht te hebben in de performance van de uitgevoerde IT-projecten. Beslissers in de organisatie leefden in de veronderstelling dat het succes van projecten in hoge mate afhing van een aantal 'helden' in de organisatie, dat te veel projecten te duur waren, te lang duurden of een slechte kwaliteit applicaties opleverden en dat de geleverde throughput onvoldoende was om aan de toenemende vraag van de opdrachtgever te kunnen voldoen. Hoewel Deel 2 is uitgewerkt op een meer wetenschappelijke wijze dan de andere twee delen, en daardoor wellicht een in de ogen van sommige lezers grote hoeveelheid (statistische) gegevens en afbeeldingen kan bevatten, is het mijn oprechte bedoeling dat ook lezers die geen specifieke training in statistiek hebben gevolgd dit deel goed kunnen lezen en de achterliggende relevantie van het onderzoek kunnen begrijpen.

Deel 3 is het praktische deel van dit boek. De doelstelling van dit deel is om zowel het IT-management, de opdrachtgevers en beslissers als de meet- en analyse-professionals van een softwareontwikkelingsorganisatie te ondersteunen bij het maken van een volgende stap om een professionele, *agile* meet- en analyse-competentie in te voeren in hun eigen omgeving. Dit deel bouwt verder op de in Deel 1 beschreven aanzet voor een standaardaanpak voor het meten en analyseren van IT-projecten, en bevat rapportagesjablonen, raamwerken en praktische voorbeelden die nuttig kunnen zijn wanneer meten en analyseren wordt ingevoerd in

een softwareontwikkelingsorganisatie en in de opdrachtgeverorganisatie aan de *business*kant.

In veel gevallen is de terminologie en de ondersteunende vakliteratuur op het gebied van meten en analyseren Engelstalig. In dit boek hanteer ik echter zoveel als mogelijk Nederlandse benamingen voor begrippen met betrekking tot dat vakgebied. Zo spreek ik over omvang in plaats van *size*, over meten en analyseren in plaats van *measurement & analysis* en basismetrieën in plaats van *core metrics*. In een beperkt aantal gevallen heb ik toch gekozen voor de Engelstalige benaming, omdat een goede vertaling ontbreekt of omdat dit de duidelijkheid ten goede komt. Ik gebruik bijvoorbeeld de term *time-to-market* en heb ook het begrip *agile* niet vertaald. Alle Engelstalige termen zijn ter verduidelijking in dit boek cursief opgenomen.

Dit boek gaat over informatietechnologie en de ontwikkeling en het beheer daarvan binnen grote, informatie-intensieve organisaties in Nederland. En binnen dat domein informatietechnologie richt het boek zich voornamelijk op projecten waarbij de nadruk ligt op het ontwikkelen en beheren van softwareapplicaties. Daarbij moet vooral niet gedacht worden dat alle onderzochte projecten uitsluitend software opleverden. Zoals binnen alle organisaties die zich bezighouden met informatietechnologie is in de IT-projecten in veel gevallen sprake van een situatie waarbij zowel software, hardware als middleware is opgenomen. Om de terminologie duidelijk te houden spreek ik daarom in dit boek niet over softwareprojecten of hardwareprojecten, maar over complete IT-oplossingen die aan een klant worden geleverd. Wanneer het proces om zo'n oplossing te maken min of meer *agile* is zie je dat vaak niet langer van projecten wordt gesproken, maar van releases. Daarom mag overal waar ik in dit boek de term (IT-)project hanteer ook aan releases gedacht worden.

Kortom, laten we dit positieve verhaal beginnen door te bekijken hoe binnen de oorspronkelijk onderzochte organisatie een meet- en analyseaanpak heeft geresulteerd in inzicht in de IT-performance. Welke bevindingen kwamen naar boven door het meten en welke aspecten maakten daarbij écht verschil? Wat waren de effecten van een overgang naar een *agile* softwareontwikkelingsproces? Allereerst worden daartoe in Hoofdstuk 1 de achtergronden en het startpunt van zo'n verandertraject geschetst aan de hand van vier belemmeringen die in de praktijk echte verbetering van softwareontwikkeling tegenhielden. Na het lezen van dit hoofdstuk zal het duidelijk zijn dat het allemaal draait om planning en dat een geslaagd project in lang niet alle gevallen ook betekent dat de performance van de softwareontwikkelingsorganisatie is verbeterd.

Na of tijdens het lezen van dit boek behoefte aan meer informatie? Kijk ook eens op de website bij deze uitgave: www.agilewerkt.nl

Deel 1

Agile projecten leveren betere resultaten op dan traditionele

‘Er zijn wel honderd manieren om een tuin aan te leggen.

De beste is om een tuinier in de arm te nemen.’

(Karel Čapek)

1 Het draait allemaal om planning

Als een direct gevolg van de kredietcrisis heeft een groot aantal bedrijven de afgelopen jaren allerlei maatregelen genomen om de kosten te verlagen. Een van die maatregelen was in veel gevallen het uitvoeren van een verbetertraject om binnen de gehele IT-organisatie procesmatig te gaan werken, en dan vaak volgens het *Capability Maturity Model Integration Model* (CMMI).⁴ De hoofddoelstelling van dat programma was doorgaans om op een hoger volwassenheidsniveau te gaan presteren waardoor belangrijke kostenbesparingen mogelijk zouden worden. Veel Nederlandse bedrijven hadden in de periode van 2009 tot 2012 een doelstelling om op CMMI-volwassenheidsniveau 2 of 3 te gaan opereren, en als gevolg daarvan was de verwachting natuurlijk dat de kosten die besteed werden aan softwareontwikkeling drastisch omlaag zouden gaan.

Die doelstelling had ook een grote, informatie-intensieve dienstverlenende organisatie waar ik in 2011 een onderzoek deed naar de performance van de IT-projecten. De verwachtingen waren hooggespannen toen het senior management in 2009 aankondigde dat de hele IT-organisatie binnen twee jaar op CMMI-volwassenheidsniveau 3 zou werken en men verwachtte kostenbesparingen waarbij zelfs verbeterpercentages van 15% en meer genoemd werden. Er werd flink geïnvesteerd in opleiding en training voor de medewerkers. Er werden proceseigenaren benoemd, managers werd verteld dat de bonussen voortaan gekoppeld zouden worden aan het al dan niet behalen van het gewenste volwassenheidsniveau. Het is dan ook niet verbazend dat CMMI vanaf dat moment hoog bovenaan op de managementagenda stond.

1.1 Meten en analyseren omwille van procesverbetering

Een van de in het kader van CMMI-ingevoerde procesgebieden was *Meten en Analyseren* (MA).⁵ Dit boek is voor een belangrijk deel het resultaat van de invoering van dat procesgebied meten en analyseren in een organisatie die opgenomen was in het onderzoek dat in Deel 2 van dit boek uitgebreid beschreven wordt. Het is interessant om te bekijken wat de oorspronkelijke aanleiding was voor die organisatie om te beginnen met meten en analyseren. In eerste instantie ging het erom dat het senior management meer inzicht wilde krijgen in de performance van softwareontwikkelingsprojecten. En daarnaast wilde men graag meer weten over de

kwaliteit van de binnen die projecten opgeleverde producten. Met softwareontwikkelingsprojecten worden in dit kader projecten bedoeld die betrekking hebben op het leveren van softwarematige oplossingen. Natuurlijk konden dergelijke projecten naast software ook wel degelijk onderdelen bevatten die gerelateerd waren aan hardware of middleware.

De invoering van meten en analyseren betrof daarom in feite een eerste fase van een verbetertraject. Men wilde op dat moment vooral weten in hoeverre de performance van de IT-organisatie marktconform was en welke trends die performance hadden beïnvloed. Pas in een volgende fase wilde men ingaan op de oorzaken achter de performance aan de hand van zogenaamde *root cause analysis*.⁶ Hierbij was het realiseren van een organisatie die op CMMI-volwassenheidsniveau 3 werkte een belangrijke doelstelling. Een volgende stap in de vernieuwing van de softwareontwikkelingsorganisatie was dat een nieuwe, op *Scrum*⁷ gebaseerde aanpak van softwareontwikkeling werd ingevoerd. Bij deze zogenaamde *agile* aanpak kregen de ontwikkelteams een meer zelfsturend karakter en lag de nadruk in eerste instantie meer op het creëren van toegevoegde waarde voor de opdrachtgever en minder op kostenaspecten van softwareontwikkeling.

1.2 Een verassende wending door een focus op *Scrum*

De oorspronkelijke verbeterdoelstelling van de onderzochte organisatie was dus het realiseren van een procesgedreven softwareontwikkelingsorganisatie die opereerde op CMMI-volwassenheidsniveau 3. Maar tijdens die transitieperiode gebeurde er iets onverwachts. Iets dat de softwareontwikkelingsorganisatie volledig op zijn kop zette. De CIO van de organisatie – een man die niet afkomstig was uit de IT-organisatie maar die een businessachtergrond had – besloot om de softwareontwikkeling niet langer volgens een watervalaanpak uit te voeren. In plaats van IT-projecten op een traditionele waterval manier uit te voeren besloot hij dat software voortaan op een *agile* manier zou worden opgezet. In een watervalaanpak, ook wel aangeduid met de term lineair, wordt achtereenvolgens een opstartfase, een ontwerpfase, een uitvoeringsfase en een afsluitingsfase uitgevoerd. Bij een *agile* aanpak worden meer verschillende korte iteraties van softwareontwikkeling uitgevoerd. Het softwareontwikkelingsproces is veel meer kortcyclisch van karakter. De CIO nam – nadat hij door twee interne *agile* supporters langdurig was bestookt met een lawine aan veelbelovende informatie over *Scrum* – het besluit dat alle softwareontwikkeling voortaan volgens *Scrum* zou worden gedaan. En die organisatieverandering zou in één keer, als een *big-bang* worden uitgevoerd. En die radicale keuze bleek achteraf grote gevolgen te hebben voor de performance van de IT-organisatie.

In een *Scrum*-aanpak kun je de softwareontwikkelteams omschrijven als kleine, min of meer zelfsturende teams waarbij de nadruk ligt op het creëren van toegevoegde waarde voor de opdrachtgevende businessonderdelen. Als gevolg van die focus bleek de besluitvorming meer dan in de voorgaande periode te liggen bij *time-to-market* en de kwaliteit van de opgeleverde softwareproducten, en wat minder op productiviteit en kostenvermindering. Anders gezegd, de opdrachtgevers wilden nieuwe oplossingen doorgaans veel sneller beschikbaar hebben dan in de voorgaande periode bruikbaar was. En zij leken daarbij – overigens zonder dat die consequentie expliciet werd gemaakt – geneigd te nemen met hogere kosten. Al snel na de overgang van watervalsoftwareontwikkeling naar een aanpak op basis van *Scrum* werd duidelijk dat die keuze resulteerde in indrukwekkende en beloftevolle performancecijfers. Hoewel het duidelijk werd dat een aantal randvoorwaarden voor de *Scrum*-activiteiten essentieel waren voor het realiseren van een goede performance, bleek dat de meerderheid van de projecten die als een *best practice*⁸ konden worden aangemerkt, waren gerealiseerd op basis van een *Scrum*-aanpak. Vooral met betrekking tot *time-to-market* bleek dat veel van de op een *agile* manier uitgevoerde projecten een verbetering van 20% of meer lieten zien ten opzichte van de set van projecten die volgens een watervalaanpak was uitgevoerd.

1.3 Geplande verbetering of een toevalstreffer?

Ook al waren de uiteindelijke resultaten van een *agile* aanpak van softwareontwikkeling veelbelovend, toch leek dit soms wel een beetje op een toevalstreffer wanneer je naar de uitkomsten van het onderzoek keek. Een analyse van de tussen 2008 en 2011 afgeronde projecten liet zien dat de overall productiviteit – gemeten in een zogenaamde productiviteitsindex (PI)⁹ – ondanks een grote hoeveelheid verbeteractiviteiten in eerste instantie helemaal niet verbeterde. Integendeel, de gemiddelde productiviteit daalde elke meetperiode weer iets ten opzichte van de periode daarvoor. Er waren echter wél indicaties – onder andere een verbetering van de voorspelbaarheid van zowel projectkosten als de opleverdatum en een vermindering van de standaardafwijking – die wezen op een beweging van de IT-organisatie naar een hoger niveau van volwassenheid.¹⁰ Drie jaar achter elkaar was die trend van een verslechterende productiviteit, maar verbeteringen ten opzichte van voorspelbaarheid en standaardafwijking, overduidelijk zichtbaar en leek het alsof de softwareontwikkelingsorganisatie hierop geen afdoende antwoord kon vinden. Anders gezegd, de beslissers leken eigenlijk niet écht geïnteresseerd in het veranderen van die trend. Het ventileren van slecht nieuws door meet- en analysespecialisten leek de IT-managers doorgaans weinig te inspireren tot verbeteracties. Er was dus zeker

geen sprake van een geplande verbeteringsactie onder de IT-managers. Maar de verhalen die de CIO van zijn medewerkers hoorde over *Scrum* hadden hem blijkbaar doen besluiten om de softwareontwikkelingsorganisatie in één keer radicaal te veranderen om te zien of daarmee wel een verbetering gerealiseerd kon worden. En dat was geen slechte keuze. Want al snel nadat een aantal *Scrum*-projecten waren afgerond, liet de analyse van de performancecijfers opeens wél veelbelovende resultaten zien. Hoewel de productiviteit van de afgeronde projecten – zowel lineair als *Scrum* – in eerste instantie geen grote veranderingen vertoonde met de voorgaande perioden, waren de verbeteringen van de *time-to-market* opmerkelijk.

1.4 De periode voorafgaand aan *Scrum*

Maar om te begrijpen waarom de resultaten van die *Scrum*-projecten zo opzienbarend waren is het goed om eerst eens wat nader te kijken naar de performance van de onderzochte organisatie in de periode die daaraan voorafging.¹¹ De gemiddelde performance gemeten over de drie jaar van 2008 tot 2011 was met betrekking tot de doorlooptijd van de projecten marktconform. De benodigde inspanning (*effort*) was echter 35% hoger dan het marktgemiddelde. Daar stond tegenover dat de projectkosten weer 15% lager waren dan marktconform. De kwaliteit van het ontwikkelproces (proceskwaliteit) was 11% beter dan het marktgemiddelde. Een verontrustend signaal was de negatieve trend die te zien was voor productiviteit, gemeten in een productiviteitsindex (PI). Het beeld dat uit het uitgevoerde onderzoek naar voren kwam was dat van een organisatie die – wellicht onbewust – voornamelijk stuurde op *time-to-market* en relatief weinig op kostenaspecten. Niet iets wat je zou verwachten in het midden van een kredietcrisis.

In de analyse zijn drie bepalende aspecten voor de performance van de softwareontwikkeling nader onderzocht; voorspelbaarheid, projectplanning en schaalgrootte. Uit de analyse bleek dat de voorspelbaarheid van zowel de projectkosten als de opleverdatum in 2011 sterk was verbeterd ten opzichte van 2010. De voorspelbaarheid met betrekking tot projectkosten was hoog (+4%), de voorspelbaarheid met betrekking tot de opleverdatum was echter relatief laag (+38%). Er was met betrekking tot de *time-to-market* duidelijk sprake van *underestimation*, of anders gezegd: de geplande doorlooptijd van de projecten was gemiddeld meer dan een derde korter dan de daadwerkelijke doorlooptijd. Wellicht was dit een belangrijke verklaring voor de ontevredenheid van de opdrachtgevers van de IT-organisatie. Opvallend was dat de organisatie erg goed bleek te zijn in het uitnutten van de geplande projectkosten, maar dat dit sterke punt feitelijk een zwak punt werd doordat de geplande productiviteit (gemeten in PI) elk jaar wat afnam en doordat er in

veel gevallen slechter dan het eigen gemiddelde gepland werd. Daarnaast liet het onderzoek zien dat er relatief veel (63%) kleine projecten (projectomvang kleiner dan 200 functiepunten) werden uitgevoerd, terwijl bleek dat middelgrote projecten (200 tot 600 functiepunten in omvang) een aanzienlijk betere performance lieten zien voor zowel tijd, geld als kwaliteit. Wanneer 50% van de gemeten kleine projecten gebundeld zouden zijn tot middelgrote projecten dan zou dat hebben geresulteerd in grote kostenbesparingen (-9,1% of € 5.500K), snellere opleveringen en een betere softwarekwaliteit.

Er leek met betrekking tot de performance van softwareontwikkelingsprojecten dus zeker verbeterpotentieel aanwezig ten opzicht van de markt. De verwachting was dat een nadrukkelijke planningsstrategie, gericht op (1) het optimaliseren van schaalgrootte, op (2) sturen op kosten (productiviteit) van projecten waar sturen op een opleverdatum niet noodzakelijk is en (3) op meer structureel plannen gebaseerd op ervaringscijfers, zou bijdragen aan een betere productiviteit, *time-to-market* én kwaliteit. Vanuit een vogelvluchtperspectief bekeken speelden in de organisatie drie dilemma's een belangrijke rol in de belemmering van daadwerkelijke performanceverbetering:

- 1 De experts planden geen verbetering.
- 2 Het managen van onzekerheden stond niet op de agenda.
- 3 Managers stuurden bijna uitsluitend op uitnutting van kosten.

Het resultaat van deze dilemma's was dat de planning van de projecten werd gerealiseerd, maar dat tegelijkertijd daardoor de gemiddelde performance van de IT-organisatie was verslechterd. Als een voorbode van een gedetailleerde beschrijving van het uitgevoerde onderzoek in Deel 2 van dit boek, ga ik in de volgende paragrafen wat nader in op de drie generieke dilemma's voor performanceverbetering, en het resultaat dat daaruit volgde.

1.4.1 Dilemma 1: experts plannen geen verbetering

Gedurende de periode van drie jaar waarin ik de oorspronkelijk onderzochte organisatie hielp om haar IT-performance op een hoger niveau te krijgen, heb ik met regelmaat en op een professionele manier geprobeerd om de managers die verantwoordelijk waren voor de goedkeuring van projectplannen ervan te overtuigen dat er een verandering noodzakelijk was in de planningsmethode van de IT-projecten. Ik legde de uitkomsten van benchmarkonderzoeken uit in vergaderingen van het managementteam. Dan liet ik hun zien welke trends te onderkennen waren en wat de effecten daarvan waren op de toekomstige en te verwachten performance, gebaseerd op ervaringscijfers van afgeronde projecten, planningen van nieuwe

projecten en metingen van de voorspelbaarheid. Ik assisteerde projectmanagers en experts – waaronder veel *enterprise architecten*, business-analisten en technische specialisten – gevraagd en ongevraagd bij vraagstukken met betrekking tot projectplanning en het monitoren van de projectvoortgang. Kortom, mijn belangrijkste doelstelling was om te zorgen voor realistische maar ook verbeterende projectplannen van een hoge kwaliteit. En ik was niet de enige die zich druk maakte om de kwaliteit van de projectplannen. Er waren kwaliteitsbewakingspecialisten, projectsupporters, specialisten op het gebied van inkoop, contractmanagement en leveranciersmanagement. Het was soms wat druk in de verbeterarena.

Ondanks al die inspanningen veranderde er echter maar weinig ten goede. Projectmanagers bleven – daarin ondersteund door hun eigen experts op specifieke vakgebieden – projectplannen opleveren die wanneer ze conform de planning zouden worden uitgevoerd, zouden leiden tot een slechtere performance dan de gemiddelde performance van alle afgeronde projecten. En de leden van de *project boards*, de managers, bleven doorgaan die ‘verslechterende’ planningen goed te keuren, zonder zich daarbij de vraag te stellen wat de effecten daarvan zouden zijn op de overall performance. Uiteindelijk kon ik geen andere reden voor dit gedrag bedenken dan dat het echte probleem was dat senior managers en opdrachtgevers hun experts gewoon niet vroegen om daadwerkelijk verbetering te plannen. Projectmanagers en hun experts planden simpelweg niet gericht op performanceverbetering omdat niemand dat aan hen vroeg. Iedereen in de organisatie praatte over procesverbetering. Er werd veel geld besteed aan de invoering van de CMMI-volwassenheidsniveau 2 procesgebieden. Maar als procesverbetering zo hoog op de agenda stond, waarom leek het dan zo te zijn dat echte verbetering – niet procesverbetering, maar verbetering van de performance van de *output* van de IT-projecten – een *no go area* was?

1.4.2 Dilemma 2: het managen van onzekerheden staat niet op de agenda

Wat de organisatie deed aan het managen en kwantificeren van projectrisico's was niet erg verheffend. Nu is dat niets nieuws. Douglas Hubbard laat in *How to Measure Anything* zien dat dit principe opgaat voor veel bedrijven, en zeker niet uitsluitend voor IT-organisaties. De insteek van Hubbard met betrekking tot meten en analyseren en onzekerheden is interessant. Hij definieert meten en analyseren als een kwantitatieve uitgedrukte vermindering van onzekerheid die gebaseerd is op een of meer waarnemingen.¹² En dat kwantificeren van onzekerheden gebeurde in de onderzochte organisatie niet. Het risicomanagement dat een projectmanager uitvoerde binnen een IT-project bleef doorgaans beperkt tot een globale inventarisatie van risico's en het uitvoeren van een of meer mitigerende maatregelen.

De keuzes die daarbij gemaakt werden leken vaak willekeurig. Een projectmanager stelde als onderdeel van een formeel projectplan een lijst samen van risico's die van toepassing waren op zijn of haar project. Doorgaans kwam die lijst van projectrisico's op mij nogal subjectief over. Waar de ene projectmanager de nadruk legde op specifieke *governance*-aspecten, benadrukte een ander juist de inhoudelijke, business-specifieke, problemen die binnen het project leken op te treden en ging een derde projectmanager helemaal los op alle technische problemen die hij kon bedenken. Voor elk risico gaf de projectmanager vervolgens aan of de impact daarvan volgens hem of haar hoog, gemiddeld of laag was. Wat overigens hoog, gemiddeld en laag in zo'n geval precies betekenden was eigenlijk nergens goed vastgelegd. Een risico dat door de ene projectmanager als hoog werd gezien, benoemde een ander als gemiddeld of zelfs laag. Gebaseerd op die risicoscore definieerde de projectmanager vervolgens mitigerende maatregelen waarmee de impact van zo'n risico lager zou worden. Het mag duidelijk zijn dat een dergelijke risicoaanpak niet echt meehelpt om de onzekerheden waarmee men in een IT-project te maken heeft te verminderen of op te lossen. En bovenal biedt deze risicoaanpak geen helder handvat voor het kwantificeren en managen van de onzekerheden met betrekking tot projectkosten, doorlooptijd en kwaliteit van een IT-project.

In de praktijk van alledag werden projectmanagers beoordeeld naar de mate waarin zij een project hadden afgerond binnen tijd en budget. Een, in de ogen van de organisatie, goede projectmanager rondde een project af binnen een marge van 10% onder of boven het laatst goedgekeurde projectplan. Als gevolg van deze beoordelingspraktijk, die overigens gemeengoed is binnen heel veel andere organisaties, was het niet vreemd dat projectmanagers en hun experts probeerden om een projectplan goedgekeurd te krijgen dat zo dicht mogelijk lag bij het *worst case scenario* dat zij konden bedenken. Zij namen bij het maken van een projectplan alle mogelijke onzekerheden die zij konden bedenken daarin op om maar zo zeker mogelijk te weten dat de door hen opgestelde planning ook echt gerealiseerd kon worden. Het is daarom erg belangrijk om te constateren dat projectmanagers in een dergelijke organisatie doorgaans niet vanzelfsprekend prioriteit leggen bij het realiseren van echte performanceverbeteringen. Een projectmanager gaat voor het projectsucces en de praktijk laat onomwonden zien dat dat echt een andere doelstelling is dan die van de managers en van de CIO. Tenzij beide partijen hun doelstellingen op elkaar hebben afgestemd natuurlijk. Maar de praktijk laat zien dat wens en werkelijkheid vaak niet hetzelfde zijn.

1.4.3 Dilemma 3: managers sturen op uitnutting van kosten

Een derde dilemma had te maken met de wijze waarop de organisatie gedurende een project daarop grip probeerde te houden; of zal ik zeggen krijgen? Wanneer een projectplan eenmaal was geaccordeerd door een *project board*, dan werd de voortgang van dat specifieke project doorgaans eenmaal per twee weken beoordeeld aan de hand van een zogenaamd *Project Control Report*. Dit rapport liet de voortgang van een project zien afgezet tegen de geplande (*forecast*) projectkosten in het projectplan. Daarnaast bevatte het een indicatie van de binnen het project gerealiseerde mijlpalen. De kostengegevens hadden doorgaans betrekking op de benodigde inspanning (*effort*) die was vastgelegd in de urenadministratie die onderdeel was van de formele projectadministratie. En daarnaast waren in de kosten de gegevens opgenomen uit eventuele contracten waarin de kosten van externe leveranciers vastgelegd waren. Als gevolg daarvan waren de gegevens met betrekking tot projectkosten doorgaans erg betrouwbaar en in hoge mate *up-to-date*. Voor gegevens met betrekking tot mijlpalen en doorlooptijd ging dat echter in veel mindere mate op. Het vastleggen van gegevens over doorlooptijd en gerealiseerde mijlpalen was meestal optioneel en gebeurde handmatig. Daardoor waren die data vaak minder betrouwbaar en veelal incompleet. In de praktijk stuurden de projectmanagers en de betrokkenen in de *project board* tweewekelijks op kostenbeheersing. Men wilde eigenlijk vooral weten in hoeverre de geplande kosten ook daadwerkelijk werden opgebruikt. Dit lijkt misschien een afdoende en correcte manier om de voortgang van een project te beoordelen, maar VU hoogleraar Chris Verhoef kwalificeert dit heel treffend als ‘het besturen van een vliegtuig met maar één instrument in de cockpit: een brandstofmeter’.¹³

1.5 Het resultaat: planning gerealiseerd, performance verslechterd

De constatering uit de voorgaande paragraaf dat de manier van projectbeheersing eigenlijk alleen maar bijdroeg aan het realiseren van *worst case* planningen kon, samen met het al eerder genoemde gegeven dat de gemiddelde voorspelbaarheid van projectkosten erg goed was, niet anders dan leiden tot een onfeilbaar scenario voor continue verslechtering van de overall productiviteit. Je zou denken dat een naar optimalisatie strevende IT-organisatie zou proberen om aan zo’n situatie een einde te maken. Maar dat bleek toch lastiger te zijn dan ik in eerste instantie dacht. Juist in een projectorganisatie waar deze manier van projectbesturing de normale gang van zaken is, zie je vaak dat een beperkt aantal helden de projecten ‘er doorheen sleuren’ en voor ogenschijnlijke successen zorgen. ‘Ik vertrouw op mijn

onderbuik, mijn intuïtie'. 'Ik heb wiskunde gestudeerd, dus je hoeft mij echt niet te vertellen hoe statistiek werkt'. Of 'Tja, jij zegt dat de datakwaliteit beter moet. Moet ik dan je cijfers zomaar vertrouwen? Ik gebruik toch liever onze eigen spreadsheet'. Projectmanagers werden beoordeeld op succesvol afronden van een project. In veel gevallen betekent succes in zo'n omgeving dat een *project board* decharge voor een project heeft gegeven. De echte performance van zo'n 'geslaagd' project telt dan eigenlijk niet mee. Het succescriterium is dan dat het project 'binnen tijd en budget is opgeleverd'. En de lijnmanagers werden beoordeeld op het behalen van een hoger volwassenheidsniveau. Eén keer per jaar werd door externe assessoren getoetst in hoeverre de IT-processen werden uitgevoerd volgens de CMMI-handleiding. En wanneer dat zo was dan konden de champagneflessen worden ontkurkt en waren de bonussen veilig gesteld. En in de onderzochte organisatie was het niet anders. Het grootste probleem was eigenlijk nog dat de personen die verantwoordelijk waren voor het verbeteren van de processen met betrekking tot projectplanning en projectbeheersing, in veel gevallen ook de 'helden' waren die met veel bravoure de projecten 'er doorheen trokken'. Lees een CMMI-boek en je kon niet anders dan concluderen dat volwassenheidsniveau 2 nog een lange weg te gaan was.

1.6 Maar er gloort hoop aan de horizon: *agile* werkt!

Al leek de stand van zaken zoals in de voorgaande paragrafen geschetst binnen de onderzochte organisatie bijna hopeloos, na drie jaar deed zich een onverwachte verandering voor die leidde tot verrassende uitkomsten. Metingen en analyses van afgeronde projecten die op basis van *Scrum* waren uitgevoerd, lieten al drie maanden nadat de eerste daarvan waren gestart grote verbeteringen van *time-to-market* zien. Op het einde van 2011 was de gemiddelde *time-to-market* met meer dan 20% verbeterd. Drie maanden daarna liet een aantal afgeronde *Scrum*-projecten vergelijkbare verbeteringen van de productiviteit zien, terwijl de kwaliteit van de projecten gelijk gebleven was aan de daarvoor gemeten (marktconforme) waarden. Een veelbelovende conclusie leek hier op zijn plaats. Daar waar het senior management drie jaar lang niet in staat leek te zijn om een negatieve trend met betrekking tot de overall performance te keren, werd dit bij wijze van spreken in een oogwenk bereikt door een veelomvattende verandering van het softwareontwikkelingsmodel. Doordat de verantwoordelijkheid voor de planning van de softwareontwikkeling werd neergelegd bij de ontwikkelteams – en niet langer bij het IT-management en de opdrachtgevers – was blijkbaar een optimale voedingsbodem voor radicale performanceverbetering gecreëerd.

Ik heb als citaat onder de titelpagina van Deel 1 van dit boek een spreuk van Karel Čapek opgenomen waarin hij betoogd dat je voor het onderhouden van een tuin toch het beste een vakman in de arm kunt nemen.¹⁴ Ik lees het citaat als een ode aan de vakmannen, de helden die onmisbaar en belangrijk zijn voor een project. Want hoewel dit boek de indruk zou kunnen geven dat ik louter voor de ratio ga – het gaat tenslotte over meten, analyseren en benchmarken – denk ik dat de ware betekenis van *agile* zit in begrippen als zelfsturende teams, verantwoordelijkheid durven geven en nemen en nadruk op vakmanschap. Daardoor creëer je gedragsverandering. En die gedragsverandering zou wel eens de echte drijfveer achter de betere performance die *agile* laat zien kunnen zijn. Zou het niet zo zijn dat je door te vertrouwen op je onderbuik en je intuïtie hele mooie dingen kunt maken? Blijkbaar had ook Čapek ondervonden dat die benadering waardevol en zelfs essentieel is. Ik geloof daar ook in. Maar je moet wel blijven meten en analyseren. Het is belangrijk dat je inzicht hebt in je eigen prestaties. En dat is eigenlijk wat ik met *Agile werkt* wil zeggen. Denk als een tuinman.

Mooi gezegd. Maar ook in de IT-wereld spelen ratio en non-ratio vaak een vals spelletje. Hoe gemakkelijk – of hoe moeilijk – is het om intuïtie met meten, analyseren en benchmarken te combineren? Soms lijken het wel twee kanten van een medaille, maar dan wel verschillende medailles. Om goed aansluiting te vinden met de bestuurder en de CIO, de beslissers achter de IT-projecten, ga ik daarom in het volgende hoofdstuk wat verder in op de wereld waarin zij acteren en op de (non)ratio's waarop zij sturen. Meten de *key performance indicatoren* (KPI's) waarop zij sturen wel de echte performance? Of besturen zij een schijnwereld waarin clichés en vooronderstellingen de richting bepalen?

1.7 Samenvatting

Uit onderzoek naar de performance van een grote IT-organisatie kwamen de volgende drie probleempunten naar voren:

- 1 Experts plannen niet automatisch verbetering.
- 2 Het managen van onzekerheden staat doorgaans niet op de agenda.
- 3 Managers sturen vaak uitsluitend op uitnutting van kosten.

Als gevolg van deze dilemma's werd de planning van afgeronde projecten weliswaar gerealiseerd, maar was de gemiddelde performance van de IT-organisatie verslechterd. Ondanks dat was de overall IT-performance van de onderzochte organisatie bij benadering marktconform. Door een verandering van het ontwikkelmodel van waterval naar *Scrum* werden al snel na de start van die nieuwe aanpak grote (20%) verbeteringen van *time-to-market* en productiviteit gemeten.

Trefwoordenregister

A

Agile 13, 22, 68
Agile Manifesto 61
AIM 77
Amsterdams Informatiemanagement
Model 77
Architectuur Functiepunten 50

B

Backfiring 52
Backlog 65
Balanced Scorecard 33
Basismetrieën 38, 45, 98
– analyse van 95
Basismodel voor meten, analyseren en benchmarken 141
Benchmarks 68
– onderzoeksuitgangspunten 79
Best Practices 122
Betrouwbaarheid van onderzoek 91
Bonus/malusaanpak 160
Bottom-up planning 132, 153
Brutoproduktiviteit 163
Budgetverdeling, metrieën 176
Business case, kwantificeren van 165
Business impact 41, 168

C

Capability Maturity Model Integrated 57
Capability Maturity Model Integration
Model 21
Certified Function Point Analyst 50
CFPA 50
CMMI 21, 57
Cone of uncertainty 55, 128
Core metrics 38, 98

Corrective and preventive action request 172
Correlatie 96
Correlatiecoëfficiënt 96
COSMIC Full Function Points 49
Cosmic Full Funtion Points 46
CPAR 172
Creatieve klasse 42

D

Dashboard 33
Dataverzameling 89, 92
Defect Rate 47
Deming-cycle 36
Demo 66
Deployment 63, 66, 125
Doorlooptijd 46, 52
– MA-leercyclus 38
– onderzochte organisaties 101
– per functiepunt 119
DSDM Atern 64
Duivelsdriehoek 151
Dynamic Systems Development Method 64

E

Effectiviteit, meten van 178
Effort 47
End of Project Report 148
Enhancements 164, 199
EQF 56
Estimating Quality Factor 56
Expertbegroting 84
Expertplanning 132, 153
Externe leveranciers 52

F

Faalfactoren, onderzochte organisaties 124

Fallback-scenario 170
 F/A plot 56
 Fibonacci-reeks 196, 199
 Five Core Metrics-methode 46
 Forecast 128
 Fouten in project 109
 – per functiepunt 120
 FPA 49
 FSM 49
 Functiepuntanalyse 38, 49
 Functiepunten 46, 48, 49
 – omrekenen naar SLOC 52

G

Galorath 69
 Gearing factor 52
 Gedragsverandering 42
 Geïntegreerde aanpak 159
 Geplande productiviteit van lopende
 projecten 133
 Geplande time-to-market van lopende
 projecten 134
 Goal Question Metric-methode 33, 173
 Goede presteerders 108
 Go Live 34
 Governance, meten van 178
 GQM 33

H

Het Nieuwe Werken 43
 hypothetisch-deductief model 78

I

IBRA Points 50
 IFPUG 49
 Incidenten 54
 – per functiepunt 121
 Indicatoren voor IT-performance 33
 Informatiemanagement 38
 Innovatie 144
 In-process product quality 35
 Input, meten van 176
 Inspanning 47, 52

– MA-leercyclus 38
 – onderzochte organisaties 104
 International Software Benchmarking
 Standards Group 70
 ISBSG 70
 ISO/IEC 24570 38, 49
 – onderzochte organisaties 92
 ISO-standaard voor Functional Size
 Measurement 49
 Iteraties 62
 IT-investeringen, meten van 176
 IT-leveranciersmanagement 58
 – kwantificeren van 159
 IT-performance, indicatoren voor 33
 IT-planningsproces 128
 IT-portfolio, kwantificeren van 54
 IT-portfoliomanagement 38, 162
 IT-risicomanagement 26, 39
 – kwantitatieve aanpak 166

K

Klanttevredenheid, meten van 176
 Kosten
 – onderzochte organisaties 105
 – per functiepunt 113
 Kwaliteit 47, 53
 – als performance-indicator 35
 – MA-leercyclus 38
 – onderzochte organisaties 120
 – van ontwikkelproces 16
 – van projectplan 26
 Kwantificeren
 – van business case 165
 – van delivery-modellen 61
 – van doelen 185
 – van IT-leveranciersmanagement 159
 – van IT-performance 45
 – van IT-portfolio's en projectplanningen 54

L

Lagelonenlanden 123
 Lean Six Sigma 61, 194
 Leverancierscontracten 52
 Leveranciersoffertes, analyseren van 161

Lineair 13, 22
 Lopende projecten, metriecken 177

M

MA-begroting 132
 MA-Benchmark en -Rapportage 149
 MA-Benchmark Rapport 149, 181
 MA-leercyclus 36, 131
 – basismetriecken 38
 – in de praktijk 147
 MA-manager 183
 Manpower Buildup Index, onderzochte organisaties 118
 MA-planning 154
 MA-Projectafsluiting 147
 – Rapport 148
 MA-Projectbeheersing 155
 MA-Projectplanning 132, 149
 – Rapport 181
 Mark II Function Points 46, 49
 Marktconformiteit 68
 MA-specialist 183
 MA-team 182
 Maturity levels 193
 MBI, onderzochte organisaties 118
 Mean 97
 Mean time to defect 35, 47
 Mean time to failure 35, 53
 Measurement & Analysis 15
 Measurement repository 180
 Meet- en analysecompetentie 171
 Metriecken
 – keuze van 186
 – meten van output 175
 – raamwerk voor 173
 – standaardset 45
 Modern portfolio theory 38
 Monte-Carlo-simulatie 56
 MoSCoW-principe 64
 MPT 38, 195
 MTTD 35, 47
 MTTF 35, 54

N

Negenvlakmodel 77
 NESMA 49
 Net present value 166, 179
 Netto contante waarde 166, 179
 Nettoproductiviteit 163

O

Omvang 46, 48
 – MA-leercyclus 38
 – meten in de praktijk 50
 – onderzochte organisaties 98
 Onderkennen van trends 96
 Onderzoeksmethode 77
 – zwakke punten 85
 Onderzoeksonderwerp 75
 Onderzoeksopzet 75
 Ontwikkelteam, grootte van 152
 Onzekerheid 56, 154
 Onzekerheidsmanagement 26

P

PDCA-cycle 36
 Performance-indicatoren, analyse van 113
 Performanceverbetering
 – adviezen 188
 – belemmeringen 25
 PI 47
 – onderzochte organisaties 115
 Plan-Do-Check-Act cycle 36
 Planningsaspecten 131
 Planningsmethode, onderzochte organisaties 132
 Planningspoker 84
 Planningsproces, agile ontwikkelmodellen 66
 Power Fit-techniek 97
 Prince2 64
 Prioriteitsstelling 171
 Probability 41
 Probability ranges 197
 Proceskwaliteit 35, 54
 – meten van 175
 – onderzochte organisaties 120

Procesmetrieke 177
 Procesverbetering 33, 57
 Procesvolwassenheid, meten van 178
 Product backlog 65
 Productiviteit 34, 47
 – bruto vs netto 163
 – lopende projecten 133
 – MA-leercyclus 38
 – meten van 175
 – onderzochte organisaties 113
 Productiviteitsfactor 49
 Productiviteitsindex 23, 47
 – algoritme 151
 – onderzochte organisaties 115, 134
 Productkwaliteit 35
 – incidenten 54
 – meten van 175
 Product owner 65, 84
 Programmapunten 50
 Projectadministratie 52
 Project Control Report 28, 131
 Project Initiation Document 84, 128, 154
 Projectkosten, formele projectadministratie 34
 Project Monitoring & Control 155
 Projectmonitoring, risicoaanpak 167
 Projectplanning, kwantificeren van 54
 Projectplanverbetering 25

Q

QSM 69, 181
 QSM SLIM Metrics 86, 198
 QSM SLIM Suite 69, 181
 – onderzochte organisaties 198

R

Rational Unified Process 63
 Ready fase 84
 Referentietabellen 149
 Rekenkundig gemiddelde 97
 Return on investment 166, 179
 Risicomanagement 26, 39, 166
 Risicomatrix 41
 Risicotaxatie 40
 R-kwadraat 96

ROI 166, 179
 Root cause analysis 22
 RUP 63

S

Schaalgrootte 152
 – onderzochte organisaties 135
 – optimaliseren portfolio's 164
 Scrum 22, 64
 – onderzochte organisaties 82
 Scrummaster 14, 65
 Scrum-team 65
 SEER 69, 181
 SEI Core Measures 45
 Sigma 97
 Six Sigma 60, 61, 194
 Size 46, 48
 Slechte presteerders 108
 – karakteristieken 124
 SLOC 46, 48, 51
 – functiepunten omrekenen 52
 Software Engineering Institute (SEI) 45
 Softwarefouten 109
 – per functiepunt 120
 Softwarekwaliteit 53
 Softwareontwikkelingsorganisatie, onder-
 zochte organisaties 80
 Source Lines of Code 51
 Source Lines Of Code 46
 Spreidingsdiagrammen 95
 – signalijnen 97
 Sprint backlog 65
 Sprint retrospective 172
 Sprints 65
 Standaardafwijking
 – statistische onderbouwing 96
 – van meetresultaten 91
 – verkleinen van 60
 Standaarddeviatie 96
 Standaardset van IT-metrieke 45
 Standard error 96
 Statistiek 95
 Sterrenscore 122
 Stories 84
 Storypoint 67

Story points 50
Succesfactoren 122, 127

T

Testadministratie 53
– fouten vastleggen 109
Tijdcompressie 151
– onderzochte organisaties 107
Time-boxes 62
– DSDM 64
Time-to-market 34
– lopende projecten 134
– meten van 175
– onderzochte organisaties 118
Toegevoegde waarde, meten van 179
Top-down planning 132, 154
Trade-off tussen tijd, geld en kwaliteit 151
Trendlijnen 86
– in onderzoek gebruikt 97
Trendlijn, onderzochte organisaties 80
Trends onderkennen 96

U

Uitbesteding, meten van 176
Uitnutting van kosten, sturen op 28

Uitwijkscenario 169
Underestimation 129
Urenadministratie 52
Uren per functiepunt 115
Use Case Points 50

V

Validiteit van onderzoek 91
Verbeteringsplanning 25
Verhoef, Chris 50, 193
Verkleinen van standaardafwijking 60
Volwassenheidsniveau 21, 42, 58
– niveau 2 82
Voorspelbaarheid 128
– van doorlooptijd 130
– van kosten 128

W

Waardecreatie
– meten van 179
– niet-tastbare 179
Waterval 13, 22
WBS 153
What-if scenario's 150
Work breakdown structure 132, 153

'Agile' staat voor een nieuwe manier van softwareontwikkeling en het managen, meten en analyseren van IT-projecten. Organisaties die de omslag hebben gemaakt van traditioneel naar agile, laten indrukwekkende resultaten zien. Ze werken gemiddeld 34% sneller en hun kosten liggen 27% lager dan die van organisaties die nog op de oude manier werken.

Agile werkt geeft inzicht in best practices en in de achterliggende karakteristieken van excellente projecten. Waarom doen agile projecten het beter dan andere? Wat zijn de belangrijkste succesfactoren? Daarnaast leest u in *Agile werkt* hoe u een agile manier van meten, analyseren en benchmarken van de IT-projecten in uw organisatie invoert.

Het lijkt een voor de hand liggende constatering, maar de beste manier om de performance in een agile omgeving te meten is om dat meten en analyseren ook op een agile wijze te doen. Agile werkt echt!

Hennie Huijgens werkt al ruim twintig jaar als specialist op het gebied van Measurement & Analysis van grote IT-portfolio's bij informatie-intensieve organisaties.

agilewerkt.nl



ISBN	978 90 12 58127 1
NUR	980



www.academicservice.nl