

eXamen.press

eXamen.press ist eine Reihe, die Theorie und Praxis aus allen Bereichen der Informatik für die Hochschulausbildung vermittelt.

Eckhart Hanser

Agile Prozesse: Von XP über Scrum bis MAP



Springer

Prof. Dr. Eckhart Hanser
Duale Hochschule Baden-Württemberg
(DHBW) Lörrach
Studiengang Angewandte Informatik,
Vertiefung Biosystem-Informatik
(Life Science)
Hangstraße 46-50
79539 Lörrach
Deutschland
map@dr-hanser.de

ISSN 1614-5216
ISBN 978-3-642-12312-2 e-ISBN 978-3-642-12313-9
DOI 10.1007/978-3-642-12313-9
Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2010

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

Grußwort

Agile Vorgehensweisen stellen, nun schon mehr als ein Jahrzehnt, eine sehr beliebte Alternative zu den so genannten „schwergewichtigen Vorgehensmodellen“ dar. Die Beschwerlichkeit, die man den älteren Methodiken SADT, SSADM, SA/SD, dem V-Modell 97 und auch wieder den jungen Modellen RUP mit UML und V-Modell XT nachsagt, begründen Entwickler und Anwender mit dem hohen Lernaufwand, der sich mit mehreren 100 Seiten Modelldokumentation präsentiert. Zudem wird die starke Bindung an einen Phasendurchlauf oder, wie im Falle des V-Modell XT, an eine Folge von Entscheidungspunkten, als zu starr geregelte Zeitstrukturierung beklagt. Auch wenn der Durchlauf angepasst werden kann, auch wenn Rücksprünge und Inkrementaufteilung erlaubt sind, so ist doch zumindest eine solide begründete Konfiguration zu bewältigen. Hinzu kommen umfangreich definierte Projektprodukte, Projektaktivitäten, Phasenzustände, die zu weitgehend feststehenden Zeitpunkten im Projektlebenslauf erreicht sein müssen, was ebenfalls als Einschränkung der Bewegungsfreiheit empfunden wird. An dieser Auffassung hat auch die Konfigurierbarkeit des V-Modell XT auf eine Projektdurchführungsstrategie „Agile Projekte“ nicht gerüttelt. Konfiguration, Dokumentation, Phasenregelung, Produktmuster, Methodenvorschriften stellen die Lernbereitschaft auf eine harte Probe, werfen berechnete Fragen der Implementierbarkeit auf und sehen sich daher auf allen Organisationsebenen mit Akzeptanzproblemen konfrontiert. Das betrifft auch gleich die gesamte Breite der beteiligten Rollen, wie z.B. die Systemanalytiker, die Entwickler, das Linienmanagement, das Projektmanagement und besonders auch die Anwender. Da kommt vielen die unmittelbare Einsatzbereitschaft der agilen Methoden wie gerufen.

Sichtbar wurde die agile Trendwende mit der immer noch heftig umstrittenen, extrem gekürzten Vorgehensvariante von Kent Beck, dem Extreme Programming, das, wie Eckhart Hanser wiedergibt, auf einen einfachen Satz von Prinzipien und Praktiken baut. Die Kürzung des Vorgehens besteht sowohl in der „Befreiung“ vom vorgeschriebenen Durchlauf, im Verzicht auf methodengestützte Spezifizierung, als auch in der Minimierung der Ausstattung mit Methoden, Vorlagen, Mustern. *Extreme* bezieht sich deshalb auch auf eine extrem dünne Dokumentation des Extreme Programming. Je agiler die Methodik ist, desto mehr Fragen nach Beispielen, Mustern, Anleitungen tauchten auf, und so wurde die Leichtgewichtigkeit in weiteren agilen Modellen wieder etwas methodisch angereichert. Die Gruppe der agilen

Methoden hat sich differenziert, ist gewachsen und hat dadurch an Bequemlichkeit und auch etwas an Übersichtlichkeit verloren. Der Literaturmarkt führt derzeit in erster Linie nur zu einzelnen agilen Modellen Monografien, und diese Werke sind leider im Brustton der Überzeugung erfolgreicher Projekte geschrieben worden, helfen deshalb wenig die Unübersichtlichkeit zu beheben. Das Buch von Eckhart Hanser leistet hier einen wichtigen sachlichen Beitrag zur Orientierung.

Die Vertreter der agilen Vorgehensweisen heben lobenswerter Weise stark auf den Mensch, seine Bestrebungen zur Zusammenarbeit, sein Kommunikationsverhalten und die sein Handeln leitenden, ethischen Werte ab. Das kommt in einigen der Prinzipien und im Agilen Manifest zum Ausdruck. Man darf dennoch hinterfragen, ob denn Prämissen, wie z.B. tägliche Morgenmeetings, Pair-Programming, gleich große Aufgabenpakete, tatsächlich den menschlichen Neigungen entsprechen und humanere Zusammenarbeit fördern oder doch eher ein Ideal bleiben. Es fällt auf, dass die „agilen Autoren“ immer von ihrer persönlichen positiven Projekterfahrung schwärmen, aber nicht von einer wissenschaftlich belegten Erkenntnis. Es wurde Zeit, dass diese sozialen, agilen Empfehlungen wenigstens mit derjenigen Genauigkeit untersucht werden, die sogar die agilen Entwickler ihren Programmen zugestehen, nämlich durch Testen. Für den viel komplexeren Akt des Zusammenarbeitens erwartet man einen sozialen Prototypen, eine Organisationsform unter kontrollierten Umgebungsbedingungen, an dem man sozial-wissenschaftlich nachweist, dass die besagten sozialen Prämissen x, die Zusammenarbeitsform y, am Besten geeignet sind für die Aufgabenstellung z. Eckhart Hanser hat seine Arbeit dieser Lücke gewidmet und die soziologischen Aspekte des Software-Entwickelns in seinem „Software Engineering Labor“ beobachtet. Ihm gebührt damit das Verdienst als einer der wenigen IT-Forscher mit seinem Meta Agile Process Model (MAP) eine soziologische Sicht in die Softwareentwicklung einbezogen zu haben und zwar mit sozialwissenschaftlichen Methoden, wie dem Feldversuch, der begleitenden Beobachtung und dem gruppendynamischen Experiment.

Die im „Organisationslabor“ – ich lasse mich hier begrifflich von Arbeiten des Tavistock Institute anregen – gewonnenen Erkenntnisse fließen in Verbesserungsvorschläge zum agilen Vorgehen ein. Damit kann der Projektmanager fundierte Kriterien an die Stelle der „agilen Schwärmerei“ setzen. Und das hilft zu beurteilen, ob eine Aufgabenstellung in einer agilen Form oder schwergewichtig angegangen werden soll. Die Wahrheit wird wohl, wie immer, in der Mitte liegen: Für einige Projekttypen eignen sich agile Modelle besser, für andere Projekttypen ist auf umfangreiche Vorgehensmodelle nicht zu verzichten. Eckhart Hanser öffnet mit seinem Buch den Blick für eine sachliche Einschätzung der Eignung agiler Modelle zu einem Projekttyp.

Ich freue mich, dass es wieder einem aktiven Mitglied der Fachgruppe „Vorgehensmodelle“ des Fachbereichs „Wirtschaftsinformatik“ der „Gesellschaft für Informatik“ gelungen ist, ein interessantes Buch herauszubringen und wünsche dem Buch eine gute Resonanz. Dass sich das Buch auch noch mit einem vor vier Jahren mit Professor Gerhard Chroust gestarteten Themenfokus „Soziale, kulturelle und lokationsspezifische Aspekte“ der Fachgruppe zusammenfällt, freut uns alle besonders. An dieser Stelle sei auch Herrn Heine vom Springer Verlag gedankt, dass er

immer „ein Ohr“ für die Fachgruppe hat. Es bleibt zu wünschen, dass der von Eckhart Hanser gestartete Weg des sozialen Experiments noch viel weiter gegangen wird, dass noch mehr sozietale Maßnahmen, Sozialkriterien und ihre Indikatoren beobachtet werden, dass weitere Informatiker die Soziologie einbeziehen. Vielleicht werden eines Tages die Aktivitäten, Methoden, Meilensteine im Projekt nicht mehr auf die technische Güte der Produkte reduziert, sondern von sozietalen Aktivitäten, soziologischen Methoden und sozialen Meilensteinen flankiert. Dann müssen sich auch die Softwareentwickler nicht schon wieder den Vorwurf gefallen lassen, am Anwender vorbei zu programmieren. Oder ist irgendjemand der Auffassung, dass z.B. das weitverbreitetste Vorgehensmodell der Welt, der Unified Process mit UML, fachanwenderfreundlich ist?

Reinhard Höhn

Sprecher der Fachgruppe „Vorgehensmodelle“ der Gesellschaft für Informatik
2007-2009, Mitglied des Leitungsgremiums der Fachgruppe 2010
Darmstadt im Februar 2010

Vorwort

Mit diesem Buch verfolge ich zwei Ziele: Zum einen soll ein Überblick über die aktuelle Welt der agilen Prozessmodelle gegeben werden. Die leichtgewichtigen Software-Entwicklungsmethodiken *Extreme Programming*, *Crystal* und *Scrum* werden ausführlich beschrieben und mit Hilfe meines studentischen Software-Engineering-Labors der Dualen Hochschule Baden-Württemberg Lörrach auf ihre Alltagstauglichkeit überprüft. Zum anderen fasse ich die Erfahrungen aus sechs Jahren experimentellen Software-Engineerings im studentischen Labor in dem neuen agilen Meta-Modell *MAP* zusammen, welches sich mit den „Zutaten“ beschäftigt, die ein erfolgreiches technisches Projekt benötigt. Die Grundidee ist einfach: Wenn man ein hochqualifiziertes Projektteam zusammenstellt, macht es keinen Sinn, einen Prozess in Form eines Flussdiagrammes vorzugeben, sondern es ist besser, die richtigen *Charaktere* oder *Typen* zusammenzubringen und mit ihnen die im studentischen Labor identifizierten notwendigen Projektrollen korrekt zu besetzen. Ein in diesem Sinn gut „ausbalanciertes“ Projektteam hat eine höhere Aussicht auf Erfolg! Diese Erkenntnis ist das direkte Resultat meines studentischen Labors.

Bei den aufgeführten Projektrollen wird in diesem Buch die männliche Form benutzt. Dies dient lediglich der Erhöhung der Lesbarkeit des Texts. Meine Erfahrung zeigt, dass Frauen im Projektgeschäft hervorragende Leistungen erbringen. Insbesondere Projektleiterinnen verstehen es oft, ihre Teams zu außergewöhnlichen Leistungen anzuspornen (Kap. 7).

Zielgruppe dieses Buchs

Dieses Buch wendet sich sowohl an Praktiker, die sich einen Überblick über den „Markt“ der agilen Ansätze verschaffen wollen, als auch an Studierende, die sich auf ihre Klausuren und Prüfungen im Bereich der agilen Prozessmodelle vorbereiten. Insbesondere Praktiker wird interessieren, welche agilen Praktiken sich im studentischen Labor und damit unter reproduzierbaren Laborbedingungen bewährt haben und welche nicht (Kap. 6). Viele Vermutungen aus Praxisprojekten konnten wissenschaftlich fundiert nachgewiesen werden.

Aufbau dieses Buchs – Agiles Manifest und agile Prozessmodelle

Nach einer kurzen Einführung in die Historie der Software-Prozessmodelle (Kap. 1) wird in Kap. 2 das Agile Manifest erläutert. Diese von Entwicklern initiierte „Gegenbewegung“ zu den schwergewichtigen Prozessmodellen wie Unified Process oder V-Modell begründet die bekannten agilen Prozessmodelle *Extreme Programming (XP)*, *Crystal* mit *Crystal Clear* und die agile Projektmanagementmethode *Scrum*. Die aufgezählten agilen Methodiken werden in den Kap. 3 bis 5 systematisch vorgestellt. Dabei stehen der Prozess und die vorgeschlagenen Praktiken sowie die Projektrollen und die erzeugten Artefakte im Vordergrund. Bei Extreme Programming (Kap. 3) werden insbesondere die traditionellen XP-Praktiken den „erweiterten“ Praktiken gegenübergestellt.

Das studentische Software-Engineering-Labor

In Kap. 6 wird das studentische Software-Engineering-Labor der Dualen Hochschule Baden-Württemberg Lörrach vorgestellt. Seit mittlerweile sechs Jahren werden in diesem experimentellen Software-Engineering-Labor agile Prozessmodelle und Methoden untersucht und auf ihre Praxistauglichkeit überprüft. Alle im Buch erwähnten Prozessmodelle wurden in diversen studentischen Projekten ausprobiert. Nach einer theoretischen Einweisung im Rahmen der Software-Engineering-Vorlesung mussten studentische Gruppen reale Projekte bearbeiten und dabei als Projektteam dem jeweils betrachteten Prozessmodell folgen. Die Ergebnisse, insbesondere bezüglich der Alltagstauglichkeit der gewählten agilen Methoden, sowie Erfahrungsberichte aus den unterschiedlichen Projekten finden sich in Kap. 6.

Danksagung

Dieses Buch wäre nicht ohne die Hilfe und Unterstützung lieber Menschen zustande gekommen. Ich danke meiner Familie, meiner Frau Susanne und meiner Tochter Stephanie, dass sie während der Erstellung dieses Buchs oft auf mich verzichtet haben. Meinem Sohn Felix danke ich für die aktive Teilnahme auf Kundenseite beim aktuellen studentischen Projekt. Meinen Eltern danke ich, dass ich das studieren durfte, was ich immer wollte.

Den Studierenden der Dualen Hochschule Baden-Württemberg Lörrach, die an den studentischen Projekten teilgenommen haben, danke ich für ihr Engagement und ihre lebhaftige Teilnahme. Sie mussten sich oft wie Versuchskaninchen vorkommen, insbesondere wenn sie die „hundertste“ Evaluation ausfüllen sollten. Und doch war ihr Feedback stets positiv und ermunterte mich zum Weitermachen.

Dem Organisationspsychologen Herrn Baldegger danke ich für die fruchtbaren Diskussionen und nützlichen Kommentare zu MAP, meiner neu entwickelten „Landkarte“ der Verhaltensweisen im Team.

Kein Projekt kommt ohne reale Kunden aus. Ich danke in diesem Zusammenhang der Geschäftsführung des Berufsförderungswerks Bad Wildbad (BFW), insbesondere Herrn Geschäftsführer Dings und Herrn Birk, sehr herzlich, dass sie das aktuelle studentische Projekt vor Ort beim Kunden und die Veröffentlichung der Ergebnisse ermöglicht haben. Herrn Birk und seinen Mitarbeitern in der IT danke ich für die aktive Teilnahme am Projekt, die uns allen neue interessante Erkenntnisse gebracht hat.

Natürlich müssen Ergebnisse aus dem studentischen Labor in der betrieblichen Praxis überprüft werden. Das ist nicht so einfach, da oftmals Geheimhaltungsgründe gegen eine Veröffentlichung sprechen. Ich danke deshalb der Werksleitung der Evonik Degussa GmbH, Werk Rheinfelden, insbesondere Herrn Werksleiter Dr. Vierbaum und Herrn Meijlink, dass ich die Ergebnisse der Diplomarbeit von Frau Dipl.-Ing. (BA) Olga Wolf zum Einsatz agiler Methoden in der Werks-IT verwenden durfte (Kap. 7). Mein besonderer Dank gilt Frau Wolf für ihre gelungene Arbeit, die ich mit Herrn Sehringer und Frau Reisch betreuen durfte.

Der Fachgruppe „Vorgehensmodelle“ der Gesellschaft für Informatik danke ich für die interessanten Diskussionen zum Thema schwergewichtige und leichtgewichtige Prozessmodelle.

Herrn Heine und Frau Herrmann vom Springer-Verlag danke ich für ihre Unterstützung beim Entstehen und bei der Veröffentlichung dieses Buchs. Herrn Heine danke ich insbesondere für seine ermunternde Aufforderung „auch mal ein kleineres Buch zu schreiben“. Ohne diese Motivation wäre das Buch nicht entstanden.

Rheinfelden
Februar 2010

Eckhart Hanser

Inhaltsverzeichnis

1	Software-Prozessmodelle	1
1.1	Ein Prozessmodell – was ist das?	1
1.1.1	Einteilung in Projekttypen	2
1.1.2	Schwergewichtige und leichtgewichtige Prozessmodelle	2
1.1.3	Prozessmodell vs. Vorgehensmodell	3
1.2	Das Phasenmodell	3
1.3	Das Spiralmodell	4
1.4	Moderne iterativ-inkrementelle Prozessmodelle	6
1.4.1	V-Modell	6
1.4.2	Unified Software Development Process (UP)	6
1.4.3	Agile Prozessmodelle	7
	Literatur	7
2	Das Agile Manifest	9
	Literatur	11
3	Extreme Programming (XP)	13
3.1	Die fünf Werte	14
3.1.1	Kommunikation	14
3.1.2	Einfachheit	14
3.1.3	Feedback	15
3.1.4	Mut	16
3.1.5	Respekt	16
3.2	Die 14 Prinzipien	17
3.2.1	Menschlichkeit	17
3.2.2	Wirtschaftlichkeit	17
3.2.3	Wechselseitiger Vorteil	18
3.2.4	Selbstständigkeit	18
3.2.5	Verbesserung	18
3.2.6	Vielfältigkeit	18
3.2.7	Reflexion	19
3.2.8	Fluss	19
3.2.9	Gelegenheit	19
3.2.10	Redundanz	19
3.2.11	Fehlschlag	20

3.2.12	Qualität	20
3.2.13	Kleine Schritte	20
3.2.14	Akzeptierte Verantwortung	20
3.3	Prozessschritte und traditionelle XP-Praktiken	20
3.3.1	Planung	21
3.3.2	Design der Software	27
3.3.3	Kodieren	30
3.3.4	Testen der Software	36
3.4	Erweiterte XP-Praktiken	37
3.4.1	Primärpraktiken	38
3.4.2	Folgepraktiken	40
3.4.3	Unterschiede zwischen erweiterten und traditionellen XP-Praktiken	42
3.5	Berühmte XP-Praktiken	43
3.5.1	Erstellen von User Stories	44
3.5.2	Paarweises Programmieren	44
3.5.3	Collective Code Ownership, Shared Code	44
3.5.4	Kontinuierliche Code-Integration, eine Codebasis	45
3.5.5	Kunde im Team, Einbeziehung des Kunden	45
	Literatur	45
4	Crystal und Crystal Clear	47
4.1	Teamgröße und Risiko – die Crystal-Familie	47
4.2	Die sieben Crystal-Eigenschaften	49
4.2.1	Regelmäßige Lieferung	49
4.2.2	Reflektierte Verbesserung	49
4.2.3	Verdichtete oder osmotische Kommunikation	50
4.2.4	Persönliche Sicherheit	51
4.2.5	Schwerpunkte bilden	51
4.2.6	Einfache Kontaktaufnahme mit Endanwendern	52
4.2.7	Technische Umgebung mit automatisierten Tests, Konfigurationsmanagement und regelmäßige Integrationen	52
4.3	Crystal Clear	53
4.3.1	Eigenschaften und Praktiken	53
4.3.2	Rollen im Projektteam	56
	Literatur	59
5	Scrum	61
5.1	Scrum – die Projektrollen	61
5.1.1	Product Owner	62
5.1.2	Team	63
5.1.3	ScrumMaster	65
5.1.4	Weitere Scrum-Rollen	67
5.1.5	Gefahr durch Rollenmissbrauch	67
5.2	Scrum – der Prozess	68
5.2.1	Scrum-Flow – Überblick	68
5.2.2	Sprint – Details	69

5.3	Scrum-Artefakte	73
5.3.1	Product Backlog	73
5.3.2	Sprint Backlog	75
5.3.3	Releaseplan und Burndown Chart	75
	Literatur	77
6	Experimentelles Software-Engineering im studentischen Labor . . .	79
6.1	Die Projekte im studentischen Labor	80
6.2	Randbedingungen der studentischen Sessions	82
6.3	Veränderung der klassischen Projektrollen in agilen Projekten . . .	84
6.3.1	Der Projektmanager im agilen Projekt	84
6.3.2	Der Qualitätsmanager im agilen Projekt	88
6.4	Neue Teamrolle – der Integrationsingenieur	92
6.5	Veränderung agiler Praktiken und Prozesse in der Praxis	93
6.5.1	Design von User Stories	93
6.5.2	Collective Code Ownership	94
6.5.3	Mini-Team-Größe: Sind XP-Paare erfolgreich?	94
6.5.4	Der Weg zur erfolgreichen Software-Integration	100
6.5.5	Crystal und die reflektierte Verbesserung	103
6.5.6	Scrum im studentischen Labor	107
6.5.7	Hierarchische Prozesse unter dem agilen Deckmantel . . .	115
	Literatur	117
7	MAP – Meta Agile Process Model	119
7.1	Team-Psychologie – Landkarte der Verhaltensweisen im Team	119
7.2	Das Super-Team	122
7.3	Was ist MAP?	123
7.4	MAP – die Projektrollen im Team	124
7.4.1	Kunde	125
7.4.2	Kommunikationsmanager	125
7.4.3	Integrationsingenieur	127
7.4.4	Team	128
7.4.5	Prozessverantwortlicher – der MAP-Beobachter	129
7.4.6	Projektrollen in der Landkarte der Verhaltensweisen	130
7.5	MAP Cycle – der Referenzprozess	131
7.6	MAP und Scrum	133
7.6.1	Vergleich der Rollen	133
7.6.2	Bestimmung des Product Owners	136
7.6.3	Prozess	139
7.6.4	Artefakte	141
7.6.5	MAP und Scrum – geht das?	142
7.6.6	Vorteile für Scrum-Anwender	143
7.7	MAP im regulierten Umfeld	143
7.8	MAP-Projekt in der Industrie	146
	Literatur	147