mitp

Mike Cohn

Mit einem Vorwort von Kent Beck

User Stories

für die agile Software-Entwicklung mit Scrum, XP u.a.

Vorwort von Kent Beck

Nach welchen Kriterien entscheiden Sie, was ein Softwaresystem leisten soll? Und wie vermitteln Sie diese Entscheidung den verschiedenen davon betroffenen Personen? Dieses Buch nimmt sich dieses komplizierten Problems an. Das Problem ist schwierig, weil alle Beteiligten unterschiedliche Bedürfnisse haben. Projektmanager wollen den Fortschritt verfolgen. Programmierer wollen das System implementieren. Produktmanager wollen Flexibilität. Tester wollen messen. Benutzer wollen ein für sie Nutzen bringendes System. Es ist sehr schwierig, produktive Auseinandersetzungen zwischen diesen unterschiedlichen Perspektiven zu führen, zu einer einheitlichen gemeinsamen Lösung zu gelangen, die alle unterstützen können, und diesen Balanceakt über Monate oder Jahre aufrechtzuerhalten.

Der Lösungsansatz, den Mike Cohn in seinem Buch *User Stories Applied* untersucht, ähnelt, oberflächlich betrachtet, früheren Versuchen, dieses Problem zu bewältigen – Anforderungen, Use Cases und Szenarios. Was ist daran so kompliziert? Sie schreiben auf, was Sie tun wollen und machen es dann. Die Fülle an Lösungen zeigt, dass das Problem nicht so einfach ist, wie es aussieht. Es kommt nämlich darauf an, was Sie schreiben und wann.

User Stories sind der Anfang des Prozesses, bei dem lediglich zwei Informationen aufgeschrieben werden: jedes Ziel, das vom System erfüllt werden muss, sowie die ungefähren Kosten für das Erreichen dieses Ziels. Dazu braucht es lediglich weniger Sätze, welche Ihnen Informationen liefern, die andere Ansätze nicht bieten. Indem es das Prinzip des »Last Responsible Moment« anwendet, wird das Team die meisten Details zu Funktionen erst kurz vor deren Implementierung niederschreiben.

Diese simple Zeitverschiebung hat zwei wichtige Effekte. Erstens kann das Team ganz einfach die »saftigsten« Features bereits in einem frühen Entwicklungsstadium einbauen, während die anderen Funktionen noch immer vage sind. Die automatisierten Tests, die die Details für jedes Feature festlegen, stellen sicher, dass frühe Features entsprechend den Spezifikationen weiterlaufen, während neue Features hinzugefügt werden. Zweitens werden durch die Bestimmung der Wertigkeit des Features bereits von Anfang an Prioritäten gesetzt, so dass der Leistungsumfang am Ende nicht abrupt abgebrochen wird, um das Lieferdatum zu erfüllen.

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

Dieses Buch wird durch Mike Cohns Erfahrungen mit User Stories zu einer praktischen Anleitung, wie User Stories in Ihrem Entwicklungsteam funktionieren können. Ich wünsche Ihnen klare und selbstbewusste Entwicklungen.

Kent Beck, Three Rivers Institute

Über die Fachkorrektoren

Susanne Hartkopf war bis 2007 Projektleiterin und wissenschaftliche Mitarbeiterin beim Fraunhofer Institut Experimentelles Software Engineering. Mit ihrem Wechsel zur Elektrobit Automotive GmbH übernahm sie die Rolle eines Process Quality Coaches, sammelte so und als Scrum Master umfangreiche Erfahrung im operativen Projektgeschäft und ist inzwischen verantwortlich für Project Quality im Automotive-Bereich. Ihrer Erfahrung nach sind Anforderungsänderungen ein inhärentes Merkmal von Softwareentwicklungsprojekten. Wenn dort agil vorgegangen wird, um dieser Tatsache Rechnung zu tragen, helfen User Stories, die Komplexität im Projekt zu verringern.

Kerstin Bücher war sechs Jahre als Beraterin für Wissensmanagement und Lernende Organisationen bei der Cogneon GmbH tätig, bevor Sie 2010 als Expertin für Wissensmanagement zum Software-Hersteller Elektrobit Automotive GmbH wechselte. Im agilen Projektmanagement werden Kommunikation und Wissenstransfer gefordert und gefördert. Deshalb unterstützt sie alle Aktivitäten in diese Richtung, ist selbst praktizierender Scrum Master und arbeitet daran, die Haltung, die agilen Methoden wie Scrum zugrundeliegt, auch in Bereichen außerhalb der Softwareentwicklung und über die Unternehmensgrenze hinaus zu etablieren.

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

Danksagungen

Dieses Buch hat von den Kommentaren vieler Kritiker und Rezensenten profitiert. Insbesondere möchte ich danken: Marco Abis, Dave Astels, Steve Bannerman, Steve Berczuk, Lyn Bain, Dan Brown, Laura Cohn, Ron Crocker, Ward Cunningham, Rachel Davies, Robert Ellsworth, Doris Ford, John Gilman, Sven Gorts, Deb Hartmann, Chris Leslie, Chin Keong Ling, Phlip, Keith Ray, Michele Sliger, Jeff Tatelman, Anko Tijman, Trond Wingård, Jason Yip und einer ganzen Handvoll namenloser Rezensenten.

Mein aufrichtigster Dank gilt meinen offiziellen Kritikern: Ron Jeffries, Tom Poppendieck und Bill Wake. Ron sorgte dafür, dass ich aufrichtig und geistig beweglich blieb. Tom öffnete mir die Augen für viele Ideen, die ich vorher nicht bedacht hatte. Bill hat mich bei der Stange gehalten und mit mir sein Akronym INVEST geteilt. Dieses Buch hat sehr stark von den Vorschlägen profitiert, die diese herausragenden Menschen eingebracht haben und auf deren Zusammenarbeit ich stolz bin.

Ich möchte auch Lisa Crispin, der Autorin von *Testing Extreme Programming*, danken, die mich ermutigt hat, dieses Buch zu schreiben und mir von ihren angenehmen Erfahrungen mit Addison-Wesley (USA) berichtet hat. Ohne ihren Zuspruch hätte ich wohl nie angefangen.

Die meiste Zeit in den vergangenen neun Jahren habe ich wohl mit Tod Golding diskutiert. Tod und ich stimmen öfter überein, als wir uns gewahr sind, aber ich lerne jedes Mal etwas Neues aus unseren Diskussionen. Er hat mir in den letzten Jahren sehr viel beigebracht und ich bin ihm zu großem Dank verpflichtet. Vieles in diesem Buch ist durch die Gespräche mit Tod angereichert worden.

Mein Dank gilt auch Alex Viggio und allen bei XP Denver, wo ich eine frühe Fassung vieler Ideen aus diesem Buch präsentieren durfte. Ich danke auch Mark Mosholder und J. B. Rainsberger, die mir berichtet haben, wie sie Software anstatt Karteikarten einsetzen. Mein Dank geht auch an Kenny Rubin, Mitautor von *Succeeding With Objects*, zusammen mit Adele Goldberg, deren offensichtlicher Stolz in ihrem Buch mir geholfen hat, nach einigen Jahren noch einmal mit dem Schreiben zu beginnen.

Ein herzliches Dankeschön auch an Mark und Dan Gutrich, die Begründer von Fast401k, die aufrichtige User Stories beisteuerten. Ich danke auch all meinen Kol-

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

legen bei Fast401k, wo wir auf einem guten Weg sind, unser Ziel zu erreichen, nämlich eines der besten Teams in Colorado zu werden.

Meiner Familie kann ich gar nicht genug danken, dass sie so viel Zeit ohne mich zugebracht hat. Vielen Dank an meine wundervollen Töchter und Prinzessinnen Savannah und Delaney. Besonderer Dank gebührt meiner wunderbaren Frau Laura, die so viel tut, auch wenn ich nur wenig dazutue.

Große Dankbarkeit schulde ich dem Team von Addison-Wesley (USA). Paul Petralia hat dafür gesorgt, dass die Arbeit von Anfang bis Ende Spaß gemacht hat. Michele Vincenti hat die Dinge am Laufen gehalten. Lisa Iarkowski hat mir unschätzbare Hilfe mit FrameMaker geboten. Gail Cocker sorgte dafür, dass sich meine Abbildungen sehen lassen können. Nick Radhuber hat zum Schluss alles zusammengefügt.

Und, last but not least, möchte ich Kent Beck für seinen hervorragenden Scharfsinn und seine Zeit danken sowie dafür, dass er dieses Buch in seine Kent-Beck-Signatur-Serie aufgenommen hat.

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

Einleitung

Mitte der 1990er Jahre hatte ich ständig ein schlechtes Gewissen. Ich arbeitete für ein Unternehmen, das alljährlich ein neues Unternehmen aufkaufte. Jedes Mal, wenn wir wieder ein neues Unternehmen gekauft hatten, wurde ich mit der Leitung der Softwareentwicklungsgruppe betraut. Und jede dieser Entwicklungsgruppen hatte wunderbare, umfangreiche Anforderungsdokumente. Ich fühlte mich unweigerlich schuldig, weil meine eigenen Gruppen nicht ebensolche wunderschönen Anforderungsspezifikationen hervorbrachten. Allerdings waren meine Gruppen durchweg sehr viel erfolgreicher beim *Erstellen von Software* als die Gruppen, die wir aufkauften.

Ich wusste, dass das, was wir machten, funktionierte. Dennoch plagte mich der Gedanke, dass wir, wenn wir seitenlange Spezifikationen schreiben würden, noch erfolgreicher sein könnten. Schließlich war es das, was in den Büchern und Artikeln stand, die ich zu jener Zeit las. Wenn erfolgreiche Softwareentwicklungsteams wunderbare Anforderungsdokumente schrieben, dann sollten wir das vielleicht auch tun. Aber wir hatten nie die Zeit dazu. Unsere Projekte waren stets zu wichtig und wurden zu bald benötigt, als dass wir mit Verzögerung beginnen konnten.

Da wir also nie Zeit hatten, ein schönes, langes Anforderungsdokument zu erstellen, einigten wir uns auf eine Arbeitsweise, die beinhaltete, dass wir mit unseren Usern sprachen. Anstatt alles niederzuschreiben, Dokumente hin- und herzuschieben und zu verhandeln, während die Zeit davonlief, redeten wir. Wir brachten Beispiele für Bildschirmanzeigen zu Papier, wir erstellten Modelle, programmierten ein bisschen und zeigten dann den zukünftigen Usern, was wir programmiert hatten. Mindestens einmal im Monat griffen wir uns eine repräsentative Gruppe an Usern heraus und zeigten ihnen genau, was wir programmiert hatten. Indem wir engen Kontakt zu unseren Usern hielten und ihnen stückweise unsere Fortschritte präsentierten, hatten wir einen Weg gefunden, ohne schöne Anforderungsdokumente erfolgreich zu sein.

Und dennoch fühlte ich mich unwohl, weil wir nicht so arbeiteten, wie wir es meiner Meinung nach eigentlich hätten tun sollen.

1999 kam Kent Becks kleines, alles umwälzende Buch *Extreme Programming Explained: Embrace Change* heraus. Und über Nacht war ich meine Schuldgefühle los. Hier war jemand, der sagte, dass Entwickler und Kunden lieber miteinander reden

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

sollten, anstatt zu schreiben, zu verhandeln und dann noch mehr zu schreiben. Kent Beck stellte vieles klar und zeigte mir viele neue Arbeitswege auf. Am wichtigsten war jedoch, dass er bestätigte, was ich aus eigener Erfahrung gelernt hatte.

Eine zu umfangreiche Zusammenstellung und Dokumentation von Upfront-Anforderungen kann ein Projekt auf vielerlei Weise zum Scheitern bringen. Am häufigsten ist dies der Fall, wenn das Anforderungsdokument selbst zu einem Ziel wird. Ein Anforderungsdokument sollte nur geschrieben werden, wenn es dazu beiträgt, das eigentliche Ziel, nämlich Software zu erstellen, zu erreichen.

Ein zweiter Grund, warum eine umfangreiche Zusammenstellung und Dokumentation von Upfront-Anforderungen ein Projekt scheitern lassen kann, ist die Ungenauigkeit geschriebener Sprache. Ich erinnere mich an eine Geschichte über ein Badewannenerlebnis mit einem Kind. Der Vater des Kindes hat die Badewanne gefüllt und hilft seinem Kind ins Wasser. Das Kleinkind, vielleicht zwei oder drei Jahre alt, steckt einen Zeh in das Wasser, zieht ihn schnell wieder heraus und bittet seinen Vater, das Wasser wärmer zu machen. Der Vater steckt seine Hand ins Wasser und ist erstaunt, dass das Wasser anstatt zu kalt, eher wärmer ist, als es seine Tochter gewohnt ist.

Nachdem er kurz über die Bitte nachgedacht hat, bemerkt der Vater, dass sie aneinander vorbeiredeten und mit denselben Worten zwei unterschiedliche Dinge bezeichneten. Die Bitte des Kindes, das Wasser »wärmer zu machen« würde von jedem Erwachsenen dahingehend interpretiert, »die Temperatur zu erhöhen«. Für das Kind bedeutete es jedoch lediglich, »das Wasser an die Temperatur anzugleichen, die ich warm nenne«.

Das geschriebene Wort ist ein sehr eindimensionales Medium, um Spezifikationen für etwas so Komplexes wie Software auszudrücken. Da Worte leicht missverstanden werden können, müssen wir das geschriebene Wort durch häufige Gespräche zwischen Entwicklern, Kunden und Usern ersetzen. User Stories bieten uns die Gelegenheit, gerade so viel niedergeschrieben zu haben, dass wir nichts vergessen und schätzen oder planen können, dabei aber auch die Zeit zur Kommunikation fördern.

Nachdem Sie den ersten Teil dieses Buches gelesen haben, werden Sie sich bald davon verabschieden, jedes kleine Spezifikationsdetail akribisch aufzuschreiben. Wenn Sie das Buch zu Ende gelesen haben, werden Sie alles Notwendige wissen, um einen story-gesteuerten Prozess in Ihrem Arbeitsumfeld einzuführen. Dieses Buch ist in vier Teile aufgeteilt und hat zwei Anhänge.

- Teil I: Der Einstieg – Eine Beschreibung aller Dinge, die Sie wissen müssen, um bereits heute damit beginnen zu können, Stories zu schreiben. Eine der Zielsetzungen von User Stories besteht darin, die Leute zum Reden statt zum Schreiben zu motivieren. Ziel dieses ersten Teils des Buches ist es, Sie baldmöglichst zum Kommunizieren zu bringen. Das erste Kapitel bietet einen

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898

Überblick darüber, was eine User Story ist und wie Sie diese einsetzen werden. Die nächsten Kapitel in Teil I enthalten weitere Informationen zum Schreiben von User Stories, zum Sammeln von Stories mittels Nutzerrollen, zum Schreiben von Stories, wenn Sie keinen Zugriff auf reale User haben, sowie zum Testen von User Stories. Teil I schließt mit einem Kapitel, das Richtlinien enthält, die Ihre User Stories verbessern werden.

- Teil II: Schätzen und Planen – Nach der erfolgreichen Sammlung von User Stories wird uns meistens als Erstes folgende Frage gestellt: »Wie lange dauert die Entwicklung?« In den Kapiteln in Teil II geht es darum, wie Sie Stories in Story Points schätzen, ein Release über einen Zeitraum von drei bis sechs Monaten planen, eine sich daraus ergebende Iteration detaillierter planen und letztendlich den Fortschritt messen und einschätzen, ob das Projekt nach Plan verläuft.
- Teil III: Häufig diskutierte Themen – Teil III beschreibt zunächst, wie sich Geschichten von den Anforderungsalternativen unterscheiden, beispielsweise von Use Cases, Anforderungsspezifikationen und Interaction-Design-Szenarios. Die Kapitel in Teil III betrachten die einzigartigen Vorteile von User Stories, woran zu erkennen ist, wenn etwas verkehrt läuft, und wie die agile Methode Scrum zusammen mit Stories eingesetzt werden kann. Das letzte Kapitel in Teil III geht auf eine Reihe kleinere Probleme ein, etwa ob Stories auf Karteikarten oder in ein Softwareprogamm geschrieben werden sollten und wie nichtfunktionale Anforderungen zu behandeln sind.
- Teil IV: Ein Beispiel – Ein erweitertes Beispiel, das helfen soll, die Enden zusammenzuführen. Wenn wir behaupten, dass Entwickler die Bedürfnisse der User am besten anhand von Stories verstehen, dann ist es wichtig, dieses Buch mit einer ausführlichen Geschichte zu beenden, die alle Aspekte von User Stories in einem Beispiel zusammenführt.
- Teil V: Anhänge – User Stories stammen aus dem Extreme Programming. Sie müssen sich nicht zwingend mit Extreme Programming auskennen, um dieses Buch lesen zu können. Dennoch finden Sie in Anhang A eine kurze Einführung in das Thema Extreme Programming. Anhang B enthält Antworten auf die Fragen am Ende der einzelnen Kapitel.

© des Titels »User Stories« (ISBN 978-3-8266-5898-3) 2010
by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: http://www.it-fachportal.de/5898