

eXtreme Programming in komplexen Projekten

Andreas Kornstädt, Stefan Roock

it-wps GmbH
Vogt-Kölln-Straße 30
22527 Hamburg
andreas.kornstaedt@it-wps.de
stefan.roock@it-wps.de

Abstract: eXtreme Programming (XP) lässt sich systematisch so anpassen, dass es für komplexe und lang laufende Projekte anwendbar wird. Dadurch können die positiven Eigenschaften des XP auch für große Projekte eingesetzt werden, ohne die Kontrolle über die Projekte zu verlieren.

1 Einleitung

Extreme Programming (XP) ist ein Vertreter der agilen Methoden, der auf Kundenzufriedenheit und Teamarbeit abzielt (siehe [Be00]). Managern erscheint XP mitunter als chaotisch und unkontrollierbar. Einige halten das Vorgehen nach XP sogar für unbeherrschbar und gefährlich, weil Planung und Kontrolle scheinbar ignoriert werden (siehe [Co02]). Ein Teilnehmer eines unserer Workshops beschrieb XP in diesem Sinne als „strukturierte Anarchie“. Bei XP scheint viel Freiheit mit wenig Regeln kombiniert zu sein. Das wirkt auf viele Entwickler anziehend, schreckt Manager jedoch ab.

Unsere Erfahrungen sind konträr zu dieser Sichtweise. Wenn XP an die jeweiligen Projektcharakteristika angepasst wird, ist der Entwicklungsprozess auch bei komplexen Anwendungsbereichen, vielen Entwicklern oder langen Projektlaufzeiten sehr gut zu kontrollieren (siehe [Li01], [LS02]). Bei IT Workplace Solution (www.it-wps.de) haben wir XP jeweils an die konkreten Herausforderungen des Projektes angepasst (siehe [BF01]). So haben wir eine Reihe systematischer Erweiterungen entwickelt, die wiederkehrenden Probleme bei der Anpassung von XP lösen. Wir verwenden die Anpassungen in einem zyklischen Prozess, der auf kontinuierliche Rückkopplung, Projektvorbereitung und auf ein passendes Rollenverständnis fokussiert. Die angesprochenen Fragestellungen umfassen:

- Wie geht man mit sehr vielen Anwendern unterschiedlicher Nutzergruppen um?
- Wie setzt man die Kundenrolle um, wenn ein Standardprodukt für ein anonymes Publikum entwickelt werden soll?

- Wie kann man in einem mehrjährigen Projekt bei zyklischem Vorgehen sicherstellen, dass man das Endziel erreicht?

Tabelle 1 zeigt eine Auswahl unserer XP-Projekte. Sie wurden so ausgewählt, dass sie eine möglichst große Brandbreite abdecken bezüglich Dauer, Größe (Anzahl der Entwickler und Coaches), Qualifikation der Projektmitglieder, Entwicklungsort und Front-End.

Tabelle 1: Projekt-Charakteristika

Projekt	Zeitdauer (verbraucht/ Rest)	Anzahl Entwickler/ Coaches	Qualifikationen der Projekt- mitglieder	Gegenstand
A	24/12 Monate (200/200 Personen- monate)	15/2	Sehr heterogen; die eingesetzten Technologien sind für viele Entwickler neu	Ablösung Alt-System einer Versicherung; Unterschiedliche Anwendergruppen; Multi- Channeling-System
B	6 Monate; beendet	6/0	Homogen; anfänglich kaum Anwendungs- wissen bei den Entwicklern	Neuentwicklung Anlagenbuchhaltung; hohe Anforderungen an Dokumentation für Wirtschaftsprüfer
C	9/27 Monate (40/100 Personen- monate)	10/0	Homogen	Neuentwicklung für Versorgungsunternehmen
D	6 Monate; beendet	6/0	Homogen	Neuentwicklung e-Business; Produktentwicklung
E	15 Monate; beendet	3/1	Heterogen	Neue Komponente für System einer Versicherung; erste OO-Erfahrungen für die Entwickler

2 Planung für mehrere Kunden/Anwender: Produktmanager

Ein wesentliches Problem konventioneller Entwicklungsprozesse ist die Kluft zwischen Kunden / Anwendern und Entwicklern. Um Missverständnisse in der Kommunikation zu verhindern, bindet XP die Kunden / Anwender während der Entwicklung eng ein (siehe [Be00]). In großen Projekten müssen allerdings die Interessen sehr vieler Anwender berücksichtigt werden. Das tritt in einer extremen Form bei Web-Projekten oder Software-Produkten auf, wo für ein anonymes Publikum entwickelt wird. Alle Anwender eng in den Entwicklungsprozess einzubinden ist dann nicht möglich.

In den Projekten A und C mussten die Entwickler mit mehreren Abteilungen zusammenarbeiten. Aus den Abteilungen wurden ungefiltert Anforderungen definiert, die sich teilweise widersprachen. Eine Lösung ist die Installation eines *Produktmanagers*, der die Anforderungen der unterschiedlichen Interessensgruppen aufnimmt und konsolidiert an die Entwickler weitergibt. Der Produktmanager nimmt dann die Kundenrolle im Sinne des XP wahr. Er nimmt damit am Planungsspiel teil und hat die Verantwortung für Umfang und Priorisierung der Anforderungen. Im Anschluss an das Planungsspiel erstellt der Produktmanager eine Feature-Liste für die Interessensgruppen, aus der hervorgeht, wann welche Anforderung umgesetzt wird.

Produktmanager sind bei komplexen Situationen auf Seiten der Kunden / Anwender nützlich bis notwendig. Allerdings wachsen die Anforderungen an den Produktmanager mit der Komplexität auf Kundenseite. Viele verschiedene – teilweise widersprüchliche - Anforderungen in einen sinnvollen Kompromiss münden zu lassen, ist eine nicht-triviale Aufgabe. Abstraktions-, Kommunikations- und Moderationsfähigkeiten sind die Grundvoraussetzungen für erfolgreiche Produktmanager. Darüber hinaus muss der Produktmanager ständig balancieren zwischen dickem Fell und Sensibilität: Kompromisse bedeuten immer, nicht alle Wünsche zu beachten. Das zieht partiell Unzufriedenheit bei den Anwendern nach sich – eine Situation, die wir in Kauf nehmen, um das System überhaupt entwickeln zu können. Der Produktmanager muss solche partielle Unzufriedenheit an sich abprallen lassen, darf aber für die Befindlichkeiten der Anwender nicht blind werden.

3 Langzeitplanung und Explorationsphase

Die Planungsinstrumente von XP stellen Iterationen und Releases in den Vordergrund. Im Planungsspiel schreiben Kunden und Entwickler Anforderungen in Form von Stories auf Karteikarten nieder. Für jede Story wird der Aufwand dabei auf der Grundlage des Erfahrungshintergrunds der Entwickler anhand ähnlicher Aufgaben und/oder ähnlicher Projekte mit ähnlich kenntnisreichen Entwicklern geschätzt.

Je mehr das vorliegende Projekt in punkto Ziele, Technologien und Teamzusammensetzung von vorherigen Projekten abweicht, desto schwieriger ist es, eine belastbare langfristige Schätzung abzugeben. In Projekt A sollte beispielsweise eine bestehende Altanwendung so abgelöst werden, dass das neue System „alles kann, was das alte konnte, nur besser“ – nur konnte leider niemand zuverlässig über das alte System Auskunft geben. Die Teammitglieder besaßen nur bruchstückhaftes Wissen über die Fachlichkeit und hatten keinerlei Erfahrung mit der zum Einsatz kommenden Technologie. In diesem Fall verbot der Erfahrungsmangel den alleinigen Einsatz von Story Cards. Große Projekte, für die eine Machbarkeitseinschätzung anhand eines vorgegebenen finanziellen Rahmens besonders wichtig ist, erfordern eine vorgeschaltete ein- bis dreimonatige *Explorationsphase*, in der anhand von Interviews und Prototypen so viel Wissen wie möglich zusammengetragen wird. Mit dem so gewonnenen fachlichen und technischen Wissen können dann Aufgabenpakete geschnürt werden, anhand derer das Team abschätzen kann, was zumindest in der nächsten Zeit wie schnell geleistet werden kann. Der nächste Schritt besteht darin, einen *Grob-Release-Plan* für das gesamte Projekt aufzustellen, in dem die Aufwände für jedes Release geschätzt werden.

Einige XP-Techniken finden in der Explorationsphase keine Anwendung, während auf andere wie zum Beispiel Stand-Up Meetings, Refactoring, Collective Ownership und Pair Programming nicht verzichtet werden sollte. Die ganze Explorationsphase ist letztendlich ein großes Planungsspiel für das kommende Projekt. Trotz des planenden Charakters der Explorationsphase betrachten wir sie nicht als einen Bruch des XP-Prinzips *no upfront design*, da im Gegensatz zur klassischen Entwicklung keine Details festgelegt werden.

4 Anforderungsermittlung

In XP halten die Kunden Ihre Anforderungen auf Story Cards fest. Im Gegensatz zur klassischen Anforderungsspezifikation stellen diese Story Cards aber keine bis ins Letzte detaillierten Anforderungen dar (siehe [BF01]). Gerade zu Beginn von Festpreisprojekten ist es aber von entscheidender Bedeutung, eine gute Schätzgrundlage für die langfristige Planung zu erhalten. Die Anforderungen müssen dabei präzise genug sein, um den Aufwand abschätzen zu können, aber auch nicht so starr sein, dass kein Spielraum mehr für Ideen des Entwicklungsteams bleibt.

In der Explorationsphase eines Projekts erarbeiten Kundenvertreter aller beteiligten Geschäftsbereiche zusammen mit den Entwicklern die Projektziele und die Anforderungsspezifikationen. Die Ziele umreißen grob, welche Geschäftsprozesse unterstützt und welche Aspekte verbessert werden sollen. Die Anforderungsspezifikation besteht dementsprechend aus zahlreichen Stories, welche die zukünftige Funktionalität auf höherem Detaillierungsgrad beschreiben. Diese *Business Stories* sind jedoch weniger detailliert als normale Story Cards und orientieren sich mehr an gesamten Geschäftsprozessen als an konkreten Tätigkeiten der Anwender.

Beispielsweise spezifizierten in Projekt C die 3 Geschäftsbereiche ihre Anforderungen als klassische Geschäftsprozesse: Listen von Arbeitsgegenständen samt generischer Tätigkeiten wie Anlegen, Lesen, Aktualisieren und Löschen. Auf dieser Grundlage war es dem Team allerdings nicht klar, wie es zu konkreten Anforderungen gelangen sollte. Daraufhin erstellte jeder Geschäftsbereich 35 Business Stories, die einen geschätzten Umfang von 600 Entwicklertagen besaßen. Diese Stories wurden in notwendige und optionale Stories unterteilt: Während die notwendigen Stories für die vertragliche Abnahme relevant waren, verhalfen die optionalen Stories dem Team zu einem besseren Verständnis des Gesamtkontexts, in dem das Projekt stattfand.

In Projekt C wurden die Business Stories zur vertraglichen Grundlage der Entwicklungsarbeit erhoben. Es wurde vereinbart, dass der Kunde die Stories auch noch im Verlauf des Projekts beliebig verändern konnte, solange der Gesamtaufwand dadurch unverändert blieb. Auf diese Weise musste für jede neue Anforderung eine alte Anforderung aufgegeben oder vereinfacht werden. Der Kunde erhält somit – ebenso wie auch bei normalen Story Cards – ein erhöhtes Maß an Flexibilität, ohne dass den Entwicklern Mehrarbeit aufgebürdet wird.

5 Zusammenfassung

Wir setzen XP nicht unter allen Umständen in jedem Projekt ein. Stattdessen passen wir die Entwicklungsmethode der jeweiligen Situation an, indem wir unser Repertoire um neue Methoden erweitern. Der meisten der eingesetzten Methoden betreffen den Bereich Planung und Kontrolle, weshalb wir – im Gegensatz zu vielen IT-Managern – der Meinung sind, dass ein angepasster agiler Entwicklungsprozess ideal für lange und komplexe Projekte geeignet ist.

6 Danksagung

Wir danken unseren Kollegen bei der it-wps und unseren Projektpartnern für die zahlreichen Diskussionen und Erfahrungsberichte.

Literaturverzeichnis

- [Be00] K. Beck: Extreme Programming Explained: Embrace Change, Addison-Wesley, 2000.
- [Co02] A. Cockburn: Agile Software Development, Addison-Wesley, 2002.
- [Li01] M. Lippert et al.: "XP in Complex Project Settings: Some Extensions," in (M. Marchesi et al., Hrsg.): Extreme Programming Perspectives, Addison-Wesley, 2001; S. 579-589.
- [LS02] M. Lippert, S. Roock, H. Wolf: Software entwickeln mit eXtreme Programming: Erfahrungen aus der Praxis, dpunkt Verlag, 2002.
- [BF01] K. Beck, M. Fowler: Planning Extreme Programming, Addison-Wesley, 2001.