

META AGILE PROCESS MODEL (MAP)

Prof. Dr. Eckhart Hanser, Berufsakademie (BA) Lörrach / Duale Hochschule BW, D-79539 Lörrach

Abstract

Schnell ändernde Anforderungen an modernes Software-Engineering bedingen agile Methoden des Projektmanagements: Mit der Einführung des "Meta Agile Process Model" (MAP) werden die wichtigsten "Zutaten" formuliert, die für eine erfolgreiche Software-Entwicklung im Team notwendig sind. Die Erfahrungen, die mit Studierenden der BA Lörrach im "Projektmanagement-Labor" gemacht wurden, lassen sich generell auf technische Projekte übertragen.

Das Agile Manifest

Agile Prozessmodelle, wie Extreme Programming (XP) [1], Scrum oder Crystal sind die Reaktion auf die so genannten „schwergewichtigen“ Prozess- bzw. Vorgehensmodelle, wie den Unified Process (UP)[2] von Rational (jetzt IBM) oder das V-Modell[3], das Vorgehensmodell der deutschen Bundesbehörden. Obwohl beide Ansätze sehr unterschiedlich sind – UP beleuchtet den iterativ-inkrementellen Software-Entwicklungsprozess, während das V-Modell die Sicht des Kunden auf das Software-Projekt in den Vordergrund stellt – gelten beide als sehr „dokumentenlastig“, was zumindest bei „Anfängern“ erheblichen zusätzlichen Aufwand bedeutet.

Branchen wie etwa die militärische oder pharmazeutische Software-Entwicklung, in denen Gefahr für Leib und Leben bestehen kann, können nicht auf ausführliche Dokumentation verzichten. In der Mehrzahl der Software-Projekte jedoch, sollte die Entwicklung funktionierender Software im Mittelpunkt stehen, wie Kritiker wie Ward Cunningham, Martin Fowler, Alistair Cockburn oder eben Kent Beck betonen. Dies führte 2001 zur Verkündung des „Manifesto for Agile Software Development“, dem Agilen Manifest[4], welches die Vorgenannten und 13 andere unterzeichneten. Leitsätze des mittlerweile weltweit unterstützten Agilen Manifests sind:

- Die individuellen Beteiligten und ihre Interaktion sind wichtiger als Prozesse und Werkzeuge.
- Die Ablieferung lauffähiger Software ist wichtiger als umfassende Dokumentation.
- Die Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen.
- Auf Änderungen der Anforderungen einzugehen ist wichtiger als die sture Verfolgung eines Plans.

Wie sieht der agile Prozess eines Teams aus?

Im Projektmanagment-Labor können schon seit 5 Jahren folgende Fragen unter „Laborbedingungen“ analysiert werden:

(1) Wie verändern sich die Projekt-Rollen und -Praktiken? Kommen neue Rollen hinzu?

(2) Praktische Erfahrung zeigt, dass Teams mit mehr als etwa 5 Mitgliedern sich in „Mini-Teams“ unterteilen. Wie sieht die Mitgliederstruktur einen solchen (erfolgreichen) Mini-Teams aus?

Diese Analysen sind die Grundlagen des entwickelten Meta-Prozessmodells „MAP“.

Projekt- und Qualitäts-Manager

Der „klassische“ Projektmanager führt sein Team und überwacht den Projektfortschritt. Er/sie erstellt das Projekthandbuch, welches das Projekt beschreibt, und den Projektplan für Aufwand, Zeit und Ressourcen. Der Projektmanager stellt die Weichen für ein erfolgreiches Projekt. In einem agilen Projekt ist diese klassische Projekttrolle jedoch nicht eindeutig definiert [5]. Meist kümmert er sich um folgende Aufgaben:

- Koordination des Projekts,
- Problemlösung,
- Kommunikation,
- Co-Design von User Stories (Anwendungsfällen in Spezifikationen, Abbildungen 3 und 4).

Die Rolle des Qualitätsmanagers, der im „klassischen“ Projekt die Projektqualität misst und verantwortlich ist für das Erstellen des Qualitätssicherungsplans, der Risikoanalyse und der Testpläne, ist im agilen Projekt nicht definiert. Er übernimmt aber auch andere Aufgaben gemeinsam mit dem Projektleiter, meist sind das: Problemlöser, Kommunikator und Co-Designer von User Stories.

Es ist bemerkenswert, dass sich die beide klassischen Projekttrollen annähern. Die klassische Rollenverteilung wird durchbrochen.

Design von User Stories und collective code ownership

Das Erstellen der User Stories findet i.A. kein breites Interesse bei den Team-Mitgliedern. Der Kunde wird als Team-Mitglied wenig zur Kenntnis genommen. Lediglich Projekt- und Qualitätsmanager (in ihrem neuen Rollenverständnis) und einige wenige Team-Members suchen den Kontakt zum Kunden, um maximale Qualität und Verständnis der User Stories zu erreichen. Oftmals werden fertige User Stories „interpretiert“ ohne nachzufragen. Erst wenn Entwickler-Mini-Teams Verantwortung für eine User Story übernehmen, werden sie aktiv und machen ein detailliertes Low-Level-Design.

Kollektive Verantwortung für den gesamten Code („Collective Code Ownership“) ist im Team nicht sehr beliebt. Außer Projekt- und Qualitätsmanager sind nur noch die wenigen Team-Mitglieder, die schon bei den User-Stories mitgewirkt haben, in der Lage, den gesamten Code zu verstehen – und nicht nur das vergleichsweise kleine Stück, das sie selbst entwickelt haben. Eine Bereitschaft für Collective Code Ownership (XP) ist im Projektmanagement-Labor nicht erkennbar.

Die neue Rolle des Integrations Engineer

Kontinuierliche Integration wird in den Teams des Projektmanagement-Labor i.d.R. nicht durchgehalten. Erst die Einführung der Rolle des „Integrations Engineer“ bringt Besserung. Durch die Besetzung dieser Position wird nach ca. 5 Wochen eine kontinuierliche Integration möglich. Integration ist nicht das Resultat einer Selbstorganisation des Teams, sondern das Ergebnis einer zentralen Planung des Projektmanagers. Integration durch gleichberechtigte Paare, wie in Extreme Programming gefordert, hat im BA-Projektmanagement-Labor bisher nicht zum Erfolg geführt.

Mini-Team-Größe: Sind XP-Paare erfolgreich?

Ein interessanter Parameter für ein Prozessmodell ist die Größe der „Untergruppen“ des Teams (Mini-Teams), in denen die einzelnen User Stories bearbeitet werden. Im Projektmanagement-Labor sind XP-Paare meist „instabil“. Die Größe der Mini-Teams ändert sich ständig, je nach Aufgabenstellung. Es gibt nach den Erfahrungen der Autoren keinen Grund für die Teamgröße 2, also für Paare. Bei übergreifenden

Aufgaben, z.B. bei der Vorbereitung der Integration ist auch eine Sechsergruppe keine Seltenheit. Wenn sich Paare bilden, bestehen sie oft aus einem stärkeren und einem schwächeren Partner, wo der erste die Struktur der Lösung erarbeitet, während der andere die Details erledigt.

Die Organisationspsychologie kennt 6 Verhaltensweisen von Team-Mitgliedern, die alle zu einem erfolgreichen Team gehören müssen: Kommunikatoren, Ideengeber, Fachexperten und Teamworker, die für das Mini-Team wichtig sind, sowie Problemlöser und Qualitätsprüfer, die (zusätzlich) für das gesamte Projektteam von Bedeutung sind (siehe dazu Abbildung 1 sowie [6]). Die Mischung aus diesen verschiedenen Verhaltensweisen macht den Erfolg des Teams. „Klassische“ XP-Paare hingegen fordern eher ähnliche Typen, da beide Paar-Mitglieder sich abwechseln sollen im strategischen und taktischen Handeln.

Besonders unter Männern fördert diese Konstellation Rivalitäten, die wiederum den Erfolg des Pairs gefährden. Im Rahmen des BA-IT-Labors wurden gut funktionierende, gleichberechtigte XP-Paare nur unter den Studentinnen gefunden!

Haupteigenschaften	bereit zur Kommunikation	erfahren	belastbar
visionär		Ideengeber	allg. menschl. Eigenschaften
kreativ			
kostenbewusst		Qualitätsprüfer	
kritisch/hinterfragend			Techn. Problemlöser/Troubleshooter
strukturiert	Kommunikator		
analytisch			
planend		Experte/Fachspezialist	
umsetzungsorientiert			Teamworker

projektarbeitsbezogene menschl. Eigenschaften

Abb. 1.: Verhaltensweisen im Team

Mini-Teams, in denen alle oben genannten Projekttypen vertreten sind, und die doch nicht zu groß werden (also vielleicht 4 bis 6 Mitglieder haben), gelten in der Organisationspsychologie als erfolgreich. Dies kann im Rahmen des Projektmanagement-Labor bestätigt werden. Trotzdem ist die Existenz eines Integration Engineers unabdingbar, da diese Aufgabe nur zentral bewältigt werden kann.

„Meta Agile Process Model“ (MAP)

Im Laufe der (Projekt-) Jahre kommt der Autor mehr und mehr zu der Überzeugung, dass der Ansatz der Entwicklung ständig neuer Prozessmodelle falsch ist. Es ist zwar wichtig, den Studierenden und Auszubildenden fundamentale Prozessmodelle als Projekt-„Baukasten“ zur Verfügung zu stellen. Für die Projektpraxis wichtiger aber ist, die minimalen „Zutaten“ zu benennen, die ein erfolgreiches Team und damit ein erfolgreicher Prozess benötigt. Man könnte diesen Ansatz als „Meta-Modell“ bezeichnen, also als Modell, das beschreibt, welche Elemente ein erfolgreiches Prozessmodell enthalten muss. Der Autor bezeichnet diesen Ansatz als „Meta Agile Process Model“ (MAP):

Ein erfolgreiches Projekt in einem nicht zu großen Team (bis vielleicht 20 Mitglieder) braucht nach den Erfahrungen aus dem Projektmanagement-Labor der BA Lörrach folgende Zutaten:

- Ein Kunde, der seine Aufgabe ernst nimmt, und dem Team nicht unbedingt immer, aber genügend oft zur Verfügung steht.

- Ein bis zwei gute und flexible Kommunikations-Manager, die sich Projekt- und Qualitätsmanagementaufgaben aufteilen, aber vor allem die Kommunikation im Team und mit dem Kunden sicherstellen. Sie müssen profunde Kenntnisse in grundsätzlichen „Prototypen“ von Prozessmodellen haben und sich nicht scheuen, das angewandte Prozessmodell abzuändern und anzupassen. (Das ist immer nötig!)
- Ein Integration Engineer, der allein verantwortlich, aber vom Projektmanagement überwacht, eine kontinuierliche Projekt-Integration einfordert und garantiert.
- Ein gut ausgewogenes Team mit einem guten psychologischen Prozess, in dem alle aus der Organisationspsychologie bekannten Typen in ausreichendem Maße vorkommen.

Wenn diese Bedingungen erfüllt sind, spielt das gewählte Software-Entwicklungs-Prozessmodell eine untergeordnete Rolle. Das Team wird im Rahmen der Projekt-Randbedingungen den geeigneten Prozess finden.

Literatur

- [1] Kent Beck „Extreme Programming Explained“, Addison-Wesley 2000
- [2] Ivar Jacobson, Grady Booch, James Rumbaugh, „Unified Software Development Process“, Addison-Wesley Professional, 1999
- [3] Andreas Rausch, Manfred Broy, Klaus Bergner, „Das V-Modell XT, Grundlagen, Methodik und Anwendungen“, Springer, 2006
oder
Wolfgang Dröschel, Manuela Wiemers, „Das V-Modell 97“, Oldenburg, 2000
- [4] <http://www.agilemanifesto.org>
- [5] Eckhart Hanser, Uwe Brecht, Andreas Michelbach, „Agile Software-Entwicklung: Extreme Programming im IT-Labor der BA Lörrach“, BA Dialog Nr. 6, 2006
- [6] Rainer Baldegger „Erfolgreich im Team“, Baldegger Verlag 2004
oder
Rainer Baldegger, Linda Gotsmann, „Ganzheitliches Projektmanagement“, Baldegger Verlag 2001