

Vergleich von klassischer und agiler Entwicklungsmethodik anhand der Entwicklung eines in StateMate simulierbaren Modells eines Türsteuergeräts

Ramin Tavakoli Kolagari
DaimlerChrysler AG
Postfach 23 60
89013 Ulm

ramin.tavakoli_kolagari@daimlerchrysler.com

Alexander Raschke
Universität Ulm
Fakultät für Informatik
89069 Ulm

alexander.raschke@informatik.uni-ulm.de

Zusammenfassung

Agile Methoden spielen eine immer größere Rolle in den verschiedensten Projekten. Sie stehen für eine flexible und kundennahe Systementwicklung und versprechen ein vergleichbar hohes Maß an Qualität. Das im folgenden beschriebene Experiment, das als Praktikum an der Universität Ulm im Sommersemester 2003 durchgeführt wurde, möchte Hinweise dahingehend bekommen, ob ein frühzeitiger Einsatz von agilen Elementen in einem klassisch geprägten Entwicklungsprozess für eingebettete Systeme möglich und sinnvoll ist.

1 Fragestellungen und Planung des Praktikums

Das klassische Requirements Engineering ist ausgezeichnet durch einen hohen Dokumentationsaufwand und schwergewichtige Prozesse. Betrachtet man beispielsweise XP, so wird gerade an diesem Prozessschritt der meiste *effort* eingespart. In Anbetracht der Tatsache, dass der Markt in immer kürzeren Zyklen immer bessere Qualität fordert, wurden Überlegungen gemacht, das Requirements Engineering zu verschlanken und leichtgewichtiger zu machen. Praktisch ganz darauf zu verzichten, wie es im XP-Ansatz gefordert wird, ist weder zielführend noch denkbar – vor allem in der Automobilindustrie, wo sich üblicherweise die gesamte Systementwicklung auf die Erstellung eines Lastenheftes (Systemspezifikation) beschränkt, das einem Zulieferer zur Weiterentwicklung gegeben wird.

Um einen ersten Eindruck von der Machbarkeit zu bekommen, haben wir das im folgenden beschriebene Experiment in einem Praktikum für Informatiker im Hauptstudium an der Uni Ulm durchgeführt. In diesem Experiment haben wir StateMate Modelle agil entwickeln lassen, da diese präzise sind wie aktuelle Systemspezifikationen und deren Erstellung den Studierenden mehr Spaß macht als die textueller Spezifikationsdokumente. Die speziell für

dieses Experiment entwickelte agile Modellierungsmethodik wurde verglichen mit einer klassischen Entwicklungsmethodik, die sich in Theorie und Praxis bewährt hat.

Wir wollten Hinweise finden, inwieweit die beiden unterschiedlichen Methoden die Modellierungszeit, die Qualität der Testfälle sowie die der StateMate Modelle, das Verständnis sowie die Änderbarkeit oder Wartbarkeit des Systems beeinflussen. Da sich aufgrund der zu erwartenden kleinen Stichprobe keine signifikanten Aussagen diesbezüglich machen ließen, haben wir uns neben der klar messbaren Daten (Zeit) vor allem auf die subjektiven Einschätzungen der Studierenden sowie unserer eigenen verlassen.

2 Ablauf des Experiments

Zunächst bildeten sich aus den Studierenden zwei Gruppen, eine, die in einem agilen Prozess und eine, die klassisch entwickeln sollte. Dann fingen die Gruppen an, gemäß ihrer von den Betreuern erstellten Prozessvorgaben, eine gekürzte Fassung der in [1] beschriebenen Version eines Türsteuergerätes in StateMate zu modellieren. Während dieser Entwicklungsarbeit bekamen beide Gruppen von den Betreuern in zeitlichem Abstand zwei Änderungswünsche, die in der Entwicklung zu berücksichtigen waren. Nach Abschluss der Entwicklungsarbeiten wurden die Modelle über die Gruppengrenze hinweg getauscht und die jeweilige Gruppe hatte nun einen letzten Änderungswunsch in das durch einen anderen Entwicklungsprozess erstellte Modell einzuarbeiten.

Die klassische Methodik war ausgezeichnet durch einen hohen Dokumentationsaufwand, um die Qualität und Verständlichkeit des zu entwickelnden Modells zu gewährleisten. Des weiteren wurde das Vorgehen durch eine dynamische Dokumentationsmethode (Use Cases) unterstützt, aus denen auch nach Abschluss des Modells die Testfälle abgeleitet wurden. Ein Abstraktionsebenenansatz gemäß [2] sollte weiterhin den Überblick über das

Modell unterstützen und Änderungen leichter einpflegen lassen.

In Abbildung 1 ist der klassische Entwicklungsprozess schematisch dargestellt. Die Betreuer des Praktikums haben in der Rolle des Kunden eine grobe Spezifikation und Änderungswünsche als Input geliefert. Die klassisch entwickelnde Gruppe hatte zwei Rollen, die nicht an Personen gebunden

Auf Grundlage der intensiven Kommunikation zwischen dem Kunden und den Entwicklern wurden zum einen die Story Cards erstellt und zum anderen die Testfälle abgeklärt. Hierbei hat es sich bewährt, dass die Rolle des Testers und die des Customer Proxy auch von zwei verschiedenen Personen eingenommen wird, um zwei Wege zu haben, die das Wissen des Kunden in das endgültige System einbinden.

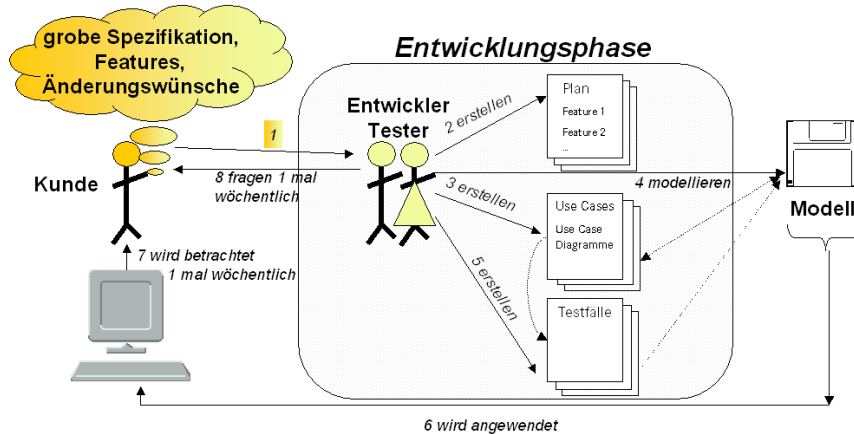


Abbildung 1. Klassischer Entwicklungsprozess.

waren: Entwickler, die die State-Mate Modelle, die Dokumentation und die Use Cases zu erstellen hatten sowie die Tester, die sich Testfälle für die Modelle auf Grundlage der Use Cases zu überlegen hatten. In wöchentlichem Rhythmus konnten Fragen an den Kunden gestellt werden sowie das Feedback des Kunden bezüglich des aktuellen Release eingeholt werden.

Die agile Entwicklungsmethodik hielt den Dokumentationsaufwand geringer, allerdings war die sorgfältige Aufbereitung der Testfälle umso umfangreicher, da das System regelmäßig mit der Entwicklung getestet wurde, so dass zu jedem Zeitpunkt lauffähige Releases zur Verfügung standen. Die Kommunikation innerhalb der Entwicklungsgruppe fand statt mit Hilfe von Story Cards, die durch einen On-Site Customer (proxy) zu erstellen waren, siehe hierzu Abbildung 2.

3 Ergebnisse

Insgesamt gibt das Experiment Hinweise darauf, dass die Qualität der Modelle und die Verständlichkeit des Systems im agilen Prozess größer als im klassischen Vorgehen ist. Auch die Zufriedenheit und innere Motivation der Gruppenmitglieder war beim agilen Vorgehen größer, was insgesamt Hinweise darauf gibt, dass die Anwendung von leichtgewichtigem Vorgehen sich beispielsweise auch im Requirements Engineering anbietet..

Literatur

- [1] F. Houdek, B. Paech. *Das Türsteuergerät – eine Beispielspezifikation*. Technischer Bericht, IESE-Report Nr. 002.02/D, Fraunhofer Institut Experimentelles Software Engineering (IESE), 2002.
- [2] H. Omasreiter, R. Tavakoli Kolagari. *Ziel und kundenorientierte Anforderungserstellung mit Abstraktionsebenen als zentraler Erfolgsfaktor bei der Entwicklung von KFZ-Software*. Proceedings der Konferenzen CONQUEST, Automation Days und EUROMOTIVE, ASQF, 2003.

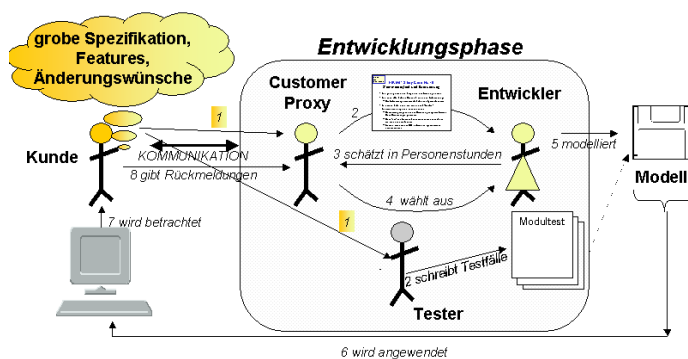


Abbildung 2. Agiler Entwicklungsprozess.