



Duale Hochschule Baden-Württemberg
Mannheim

Projektarbeit 1

Optimierung des Release-Prozesses bei SAP TwoGo mithilfe von Continuous Delivery

Studiengang Wirtschaftsinformatik Software Engineering

Bearbeitungszeitraum: 08.08.2013 bis 08.11.2013

Verfasser
Matrikelnummer

Johannes Haaß
4101368

Kurs
Studiengangsleiter

WISE12B
Prof. Dr. Thomas Holey

Ausbildungsfirma

SAP AG
Dietmar-Hopp-Allee 16
Walldorf

Firmenbetreuer

Dirk Lehmann
dirk.lehmann@sap.com

Wissenschaftlicher Betreuer

Tobias Kißmer
tobias.kissmer@schaeffler.com

Abstract

Mitfahrgelegenheiten erleben momentan eine starke Nachfrage. Besonders kleine Unternehmen und Start-ups haben diesen Trend erkannt und entsprechende Cloud-Lösungen auf den Markt gebracht. Durch schlanke Strukturen und flache Hierarchien können diese sehr schnell neue Funktionen an den Kunden bringen. Große Unternehmen, wie beispielsweise SAP, können durch ihre Größe dieses Tempo nicht mitgehen und sind damit im Bereich der Cloud-Lösungen im Nachteil.

Im Rahmen dieser Projektarbeit werden Optimierungen durchgeführt, um diesen Nachteil auszugleichen. Dazu wird der Release-Prozess der SAP Mitfahrlösung TwoGo analysiert und den zentralen Merkmalen von Continuous Delivery gegenübergestellt. Aus dieser Gegenüberstellung werden potentielle Optimierungsmöglichkeiten für den Release-Prozess von SAP TwoGo abgeleitet. Diese betreffen sowohl technische als auch organisatorische Aspekte.

Eine konkrete Optimierungsmöglichkeit ist der Deployment-Test. Dieser überprüft den Deployment-Vorgang und erkennt dabei Fehler, Warnungen und Erfolge. Zur Umsetzung dieses Deployment-Tests wird der Deployment-Vorgang analysiert, ein Entwurf erstellt und dieser implementiert und getestet.

Inhaltsverzeichnis

	Seite
Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
1 Einleitung	1
1.1 Problemstellung und -abgrenzung	1
1.2 Ziel der Arbeit und Vorgehensweise	1
2 Planung und Vorgehen	2
2.1 Projektmanagement	2
2.1.1 Anforderungen	2
2.1.2 Einrichtung von Trello und GitHub	3
2.1.3 Praktische Erfahrungen mit Trello und GitHub	3
2.2 Iterationen	4
2.2.1 Erstellung des ersten Fachkonzepts	4
2.2.2 1. Iteration	5
2.2.3 2. Iteration	5
2.2.4 3. Iteration	6
3 Anforderungsspezifikation	7
3.1 Zielbestimmung	7
3.1.1 Muss-Kriterien	7
3.1.2 Kann-Kriterien	7
3.1.3 Abgrenzungskriterien	8
3.2 Einsatz	8
3.3 Umgebung	8
3.4 Funktionalität	8
3.5 Daten	9
3.6 Leistungen	9
3.7 Benutzeroberfläche	9
3.8 Qualitätsziele	9
4 Vorgehen und Iterationen	10

5 Zusammenfassung und Ausblick	11
6 Planung und Vorgehen	12
6.1 Projektmanagement	12
6.1.1 Anforderungen	12
6.1.2 Einrichtung von Trello und GitHub	13
6.1.3 Praktische Erfahrungen mit Trello und GitHub	13
7 Planung und Vorgehen	15
7.1 Projektmanagement	15
7.1.1 Anforderungen	15
7.1.2 Einrichtung von Trello und GitHub	16
7.1.3 Praktische Erfahrungen mit Trello und GitHub	16
8 Planung und Vorgehen	18
8.1 Projektmanagement	18
8.1.1 Anforderungen	18
8.1.2 Einrichtung von Trello und GitHub	19
8.1.3 Praktische Erfahrungen mit Trello und GitHub	19
Literaturverzeichnis	i
Anhang	ii
Ehrenwörtliche Erklärung	iii

Abkürzungsverzeichnis

CD Continuous Delivery

Abbildungsverzeichnis

1 Einleitung

1.1 Problemstellung und -abgrenzung

1.2 Ziel der Arbeit und Vorgehensweise

2 Planung und Vorgehen

Bei der Planung und Entwicklung von Avalon soll Effektivität und Zielstrebigkeit eine wichtige Rolle spielen. Deshalb war es uns von Beginn an wichtig, genügend Zeit in die Planung zu investieren. Während der ersten Planungsphase haben wir entschieden, das Projektmanagement softwareseitig zu unterstützen um die Organisation und die Zusammenarbeit untereinander zu vereinfachen sowie den zeitlichen Aufwand für die Organisation zu minimieren. Außerdem haben wir entschieden Iterationen bei der Entwicklung durchzuführen, damit Fehler leichter aufgedeckt werden und Prioritäten für zentrale Aspekte von Avalon gesetzt werden können.

2.1 Projektmanagement

2.1.1 Anforderungen

Für die softwareseitige Unterstützung des Projektmanagements haben wir zunächst Anforderungen herausgearbeitet. Für uns ergaben sich drei zentrale Anforderungen an diese Software.

Die erste Anforderung ist es einen Gesamtüberblick über den aktuellen Stand des Projektes darzustellen. Dazu soll eine Aufgabenliste angezeigt werden, bei der sich Aufgaben auch an einzelne Personen zuordnen lassen.

Die zweite Anforderung ist eine zentrale Codeverwaltung. Damit soll erreicht werden das die Teammitglieder immer Zugriff auf den aktuellen Code haben und Erkennbar ist wer welche Veränderungen vorgenommen hat. Zudem soll die Codeverwaltung auch eine Versionsverwaltung bieten um gegen Datenverluste jeglicher Art vorzubeugen.

Die dritte Anforderung bezieht sich auf die Anforderungen 1 und 2. Da wir oft von unterschiedlichen Orten und Rechnern arbeiten, sollen die Daten überall und jederzeit verfügbar sein. Um diese Anforderung zu erfüllen lag es nahe, eine Lösung zu nutzen, bei der die Daten, online, in einer Cloud gespeichert werden. Aufgrund von persönlichen Erfahrungen aus vergangenen Projekten und Internet

Recherchen haben wir entschieden, dass wir für den Gesamtüberblick *Trello* benutzen und für die Codeverwaltung *GitHub* benutzen. Beide Tools erfüllen die oben genannten Anforderungen.

2.1.2 Einrichtung von Trello und GitHub

Die Einrichtung von Trello ist sehr einfach, da keine Software installiert werden muss und alles im Browser läuft. Zur Einrichtung ist nur eine Registrierung und die Erstellung eines gemeinsamen *Boards* notwendig. Ein solches Board besteht aus beliebig vielen Spalten, in denen die konkreten Aufgaben stehen. Unser Avalon Board hat folgenden Aufbau:

To-Do	Doing	Done	Idea's
Kaufalg.	Use-Cases	Marktanalyse	Historie
...

Auch die Einrichtung von GitHub ist recht einfach. Auch hier müssen sich die Mitglieder auf der GitHub Homepage registrieren. Danach erstellt ein Mitglied ein neues Repository und fügt die anderen Mitarbeiter als Collaborators hinzu. Zur Nutzung ist es noch notwendig *Git* oder die *GitHub-GUI* auf den einzelnen Rechnern zu installieren. Dieses Tool benötigt man um das aktuelle Repository herunterzuladen und Änderungen als Commit hochzuladen. Da in unserem Team nur sehr geringe Erfahrungen mit Git vorhanden waren, entschieden wir uns nur die GitHub-GUI zu nutzen.

2.1.3 Praktische Erfahrungen mit Trello und GitHub

Während Entwicklung von Avalon stellte sich besonders Trello als sehr hilfreiches, einfaches und fehlerfreies Tool dar. Die Funktionsweise wurde jedem sofort verständlich und war selbst erklärend. Mit diesen Eigenschaften konnte Trello immer einen guten Überblick über den aktuellen Stand liefern und erfüllte die von uns gestellten Anforderungen sehr gut.

GitHub dagegen lief nicht ganz so unproblematisch. Die Funktionsweise über die GUI war zwar sehr logisch und einleuchtend, allerdings kam es regelmäßig zu teils merkwürdigen Fehlern. Das größte Problem entstand, wenn zwei Mitglieder die gleiche Datei bearbeiteten. Das Mergen der Datei über die GUI war nie möglich und manuelles Mergen über die Konsole funktionierte nur selten. Um dieses Problem schnell und einfach zu lösen, hat einer der beiden seine Änderungen separat gespeichert und das Repository neu heruntergeladen. Diese Fehler führte oft zu Frustrationen innerhalb des Teams. Der Grund für diese Fehler lag sowohl an unsere Unerfahrenheit mit Codeverwaltung und an der GitHub GUI. Ein weiterer frustrierender Fehler war das Überschreiben von einigen Dateien mit älterem Inhalt. All diese Probleme führten zu Überlegungen GitHub durch ein anderes Tool, wie Dropbox, zu ersetzen. Diese Überlegungen wurden jedoch wieder verworfen, da damit vermutlich noch weitere Probleme entstanden wären. Im Nachhinein wäre es besser gewesen die Git-Bash zu verwenden und eine ausführliches Tutorial zu Git zu machen.

2.2 Iterationen

Aufgrund der Zeitspanne von drei Monaten über welche dieses Projekt verlief, haben wir uns entschieden zunächst ein Fachkonzept zu erstellen und darauf aufbauend drei Iterationen durchzuführen. Nach den Prinzipien der agilen Softwaremethodik sind die Übergänge zwischen diesen Iterationen nach beide Richtungen offen. Dies war für uns sehr essentiell, um Ergänzungen vorzunehmen und bestimmte Aspekte im Nachhinein noch konkretisieren zu können.

2.2.1 Erstellung des ersten Fachkonzepts

Bevor wir mit den Iterationen beginnen konnten, war es zunächst notwendig das Fachkonzept zu planen und zu erstellen. Als Grundlage für diese wurde zunächst eine Marktanalyse durchgeführt um die zentralen Merkmale des Smartphone-Marktes herauszuarbeiten. Hierbei spielten auch unsere eigene Erfahrungen mit Smartphones im Unternehmen und im privaten eine entscheidende Rolle. Aus diesen Erkenntnissen konnte ein erstes Modell unseres Spiel skizziert werden

und die grundlegende Spielidee festgemacht werden. Diese Erkenntnisse wurde in Form eines Pflichtenheft festgehalten und führten zur Erstellung der ersten Use-Cases.

Aus dem Pflichtenheft und den Use-Cases ließen sich die ersten Klassen identifizieren. Dabei wurden nach den Regeln der Objekt Orientieren Analyse nur Klassen identifiziert, die für die Logik des späteren Spiels notwendig sind. Während dieser Phase ließen sich auch schon einige Assoziationen und Vererbungsstrukturen, wie zum Beispiel bei den Departments, erkennen. Mit diesem grundlegenden Fachkonzept war es möglich den ersten Iterationsschritt einzuleiten. Während den Iterationen wurde das Fachkonzept an einigen Stellen noch um einige Klassen, Assoziationen und Vererbungen ergänzt und präzisiert.

Zeitlich wurde das erste Fachkonzept innerhalb der ersten zwei Wochen des Projektes erstellt.

2.2.2 1. Iteration

Die erste Iteration hatte das Ziel die zentralen Merkmale des Fachkonzept zu implementieren. Dazu wurden die wichtigsten Klassen aus dem Fachkonzept, wie zum Beispiel die Departments in Java programmiert. Während dieser Tätigkeit sind uns immer wieder einige Lücken und Fehler im Fachkonzept aufgefallen, die ein funktionierendes Spiel verhindert haben. Um die Vollständigkeit und die Richtigkeit zu gewährleisten haben wir in diesen Fällen das Fachkonzept um die fehlenden Merkmale ergänzt beziehungsweise die Fehler behoben. Um nicht den Überblick zu verlieren Parallel zur Implementierung der Klassen des Fachkonzepts wurden bereits, soweit wie möglich, die ersten Unit-Tests geschrieben um zu gewährleisten, dass der Quellcode fehlerfrei ist. Nachdem die wichtigsten Klassen implementiert waren, wurde der zweite Iterationsschritt eingeleitet.

2.2.3 2. Iteration

Im zweiten Iterationsschritt wurden weitere Klassen aus dem Fachkonzept implementiert. Besonders wurde die Implementierung des Market und der Consumer Groups priorisiert, da diese ein sehr wichtiges Element von Avalon sind. Während dieser Phase wurde auch die Excel-Tabelle erstellt, welche den Kaufalgorithmus

simuliert. Damit konnte die Funktionsweise des Algorithmus-es bereits vor der Implementierung sichergestellt werden. Ein zweiter wichtiger Punkt in dieser Iteration war die Implementierung des Client/Server Modells und der Config-Datei. Somit ging diese Iteration über das Fachkonzept hinaus und fügte weitere Mechanismen hinzu, die für das fertige Spiel zwingend notwendig sind. Auch während dieser Iteration wurden, soweit wie möglich, Unit-Tests geschrieben um Fehler abzufangen.

2.2.4 3. Iteration

In der dritten und letzten Iteration wurden die noch fehlenden Klassen des Fachkonzepts implementiert. Konkret handelte es sich dabei um die Zufallsereignisse. Außerdem wurde noch das User Interface implementiert und Fehler behoben.

3 Anforderungsspezifikation

3.1 Zielbestimmung

Mit diesem Unternehmensplanspiel soll ermöglicht werden, dass ein Spieler ein Unternehmen steuert, dass Smartphones produziert und diese Endkunden verkauft.

3.1.1 Muss-Kriterien

- Der Spieler kann die Abteilungen Einkauf, Produktion, Verkauf, Forschung, Marketing und Rechtsabteilung steuern und verwalten.
- Die Spieler spielen gegen andere menschliche Spieler.
- Die Nachfrage nach Smartphones soll nach realistischen Werten simuliert werden.
- Das Spiel läuft rundenbasiert ab. Eine Runde entspricht einer Periode.
- Spieler können über ein Client-Server Modell gemeinsam spielen.
- Abteilungen und Produkte können durch ein "Level up"verbessert werden.
- Es gibt einen zentralen Markt auf dem die Smartphones abgesetzt werden können.
- Zufallsereignisse sollen den Spielverlauf für einzelne Unternehmen und für den ganzen Markt beeinflussen.

3.1.2 Kann-Kriterien

- Die Spieler können über einen Chat miteinander kommunizieren.
- Downgrade von Abteilungen um Fixkosten zu senken.

3.1.3 Abgrenzungskriterien

Aufgabe des Spieles ist es nicht sämtliche Aspekte eines Unternehmens zu simulieren. Es genügt wenn wesentliche Merkmale eines Smartphones-Herstellers abgebildet werden. Es wird unter anderem keinen Wert darauf gelegt Mitarbeiter zu simulieren und das Lager zu vergrößern. Eine Künstliche Intelligenz um Einzelspiele zu spielen ist nicht vorgesehen.

3.2 Einsatz

Für dieses Unternehmensplanspiel gibt es keine konkrete Zielgruppe. Es soll aber von allen Interessenten verwendet werden können, die sich für ein wirtschaftliches Planspiel begeistern können.

Der Anwendungsbereich spielt keine Rolle. Da die Software regelmäßig Eingaben benötigt, ist ein unbeaufsichtigter Betrieb nicht vorgesehen.

3.3 Umgebung

Die Software soll auf Windows-Rechnern laufen, auf denen eine aktuelle Java Version installiert ist. Die Hardware muss den Java Mindestanforderungen entsprechen und es muss eine Netzwerkkarte vorhanden sein. Um das Spiel zu spielen sollen mindestens zwei Spieler erforderlich sein.

3.4 Funktionalität

In diesem Unternehmensplanspiel sind die folgenden Abläufe vorgesehen:

- Das Unternehmen kauft Rohstoffen ein.
- Das Produzieren von Smartphones.
- Das Verkaufen von Smartphones auf einem gemeinsamen Markt.
- Das Festlegen der Verkaufspreise für Smartphones.

- Das Upgraden von Abteilungen um die Kapazität bzw. Erfolgchance zu verbessern.
- Das Starten von unterschiedlichen Forschungskampagnen mit unterschiedlichen Auswirkungen.
- Das Ausspionieren anderer Spieler um bessere Smartphones zu produzieren.
- Das Starten von Marketingkampagnen um das Image zu verbessern.
- Das Überprüfen und Verklagen von anderen Spieler um sich gegen Spionagen zu wehren.
- Das Erstellen von Patenten um eine besser Position bei Rechtsstreiten zu haben.
- Das Anfechten eines Rechtsstreites.

3.5 Daten

In diesem Planspiel müssen die Benutzereingaben nicht dauerhaft gespeichert werden. Die Startparameter für das Spiel sollen in einer Datei dauerhaft gespeichert werden.

3.6 Leistungen

Es gibt keine konkreten Vorgaben bezüglich der Leistung. Der Nutzer soll aber eine positive Nutzererlebnis haben. Dazu soll das Spiel zügig auf Eingaben reagieren. Insgesamt sollten Wartezeiten unter 10 Sekunden liegen.

3.7 Benutzeroberfläche

Die Benutzeroberfläche dieses Planspiels soll alle veränderbaren und nicht veränderbaren Parameter des Unternehmens darstellen. Zudem soll die Oberfläche

minimalistisch und übersichtlich sein. Ein ansprechendes Design ist kein entscheidendes Kriterium.

3.8 Qualitätsziele

Das Unternehmensplanspiel soll am Ende fehlerfrei laufen und soll den hier genannten Anforderungen entsprechen. Der Code soll übersichtlich und kommentiert sein.

4 Vorgehen und Iterationen

5 Zusammenfassung und Ausblick

6 Planung und Vorgehen

Bei der Planung und Entwicklung von Avalon soll Effektivität und Zielstrebigkeit eine wichtige Rolle spielen. Deshalb war es uns von Beginn an wichtig, genügend Zeit in die Planung zu investieren. Während der ersten Planungsphase haben wir entschieden, das Projektmanagement softwareseitig zu unterstützen um die Organisation und die Zusammenarbeit untereinander zu vereinfachen sowie den zeitlichen Aufwand für die Organisation zu minimieren. Außerdem haben wir entschieden Iterationen bei der Entwicklung durchzuführen, da damit Fehler leichter aufgedeckt werden und Prioritäten für wichtige Aspekte gesetzt werden können.

6.1 Projektmanagement

6.1.1 Anforderungen

Da es sich bei Avalon um ein sehr komplexes Projekt handelt, soll Zur Unterstützung des Entwicklungsprozesses haben wir uns entschieden Programme einzusetzen, die Aufgaben des Projektmanagements abnehmen bzw. erleichtern. Für uns ergaben sich zwei zentrale Anforderungen an diese Programme. Die erste Anforderung ist es einen Gesamtüberblick über den aktuellen Stand des Projektes darzustellen. Dazu gehört, was noch erledigt werden muss, was gerade gemacht wird und was bereits erledigt wurde. Die zweite Anforderung ist eine zentrale Codeverwaltung mit Versionsverwaltung, um das gemeinsame und zeitgleiche Programmierung zu ermöglichen. Außerdem soll durch Versionsverwaltung gegen Datenverluste vorgebeugt werden.

Da wir oft von unterschiedlichen Orten und Rechnern arbeiten, müssen die genannten Anforderungen immer und überall erfüllt werden. Um diese, weitere Anforderung zu erfüllen lag es nahe, die Daten in der Cloud zu speichern. Aufgrund von persönlichen Erfahrungen und Internet Recherchen haben wir

entschieden, dass wir für den Gesamtüberblick *Trello* benutzen und für die Codeverwaltung *GitHub* benutzen. Beide Tools erfüllen die oben genannten Anforderungen.

6.1.2 Einrichtung von Trello und GitHub

Die Einrichtung von Trello ist sehr einfach, da keine Software installiert werden muss und alles im Browser läuft. Zur Einrichtung ist nur eine Registrierung und die Erstellung eines gemeinsamen *Boards* notwendig. Ein solches Board besteht aus beliebig vielen Spalten, in denen die konkreten Aufgaben stehen. Unser Avalon Board hat folgenden Aufbau:

To-Do	Doing	Done
Kaufalg.	Use-Cases	Marktanalyse
...

Auch die Einrichtung von GitHub ist recht einfach. Auch hier müssen sich die Mitglieder auf der GitHub Homepage registrieren. Danach erstellt ein Mitglied ein neues Repository und fügt die anderen Mitarbeiter als Collaborators hinzu. Zur Nutzung ist es noch notwendig *Git* oder die *GitHub-GUI* auf den einzelnen Rechnern zu installieren. Dieses Tool benötigt man um das Repository herunterzuladen und Änderungen hochzuladen. Da in unserem Team nur geringe Erfahrungen mit Git vorhanden waren, entschieden wir uns die GitHub-GUI zu nutzen.

6.1.3 Praktische Erfahrungen mit Trello und GitHub

Während Entwicklung von Avalon stellte sich besonders Trello als sehr hilfreiches, einfaches und fehlerfreies Tool dar. Die Funktionsweise wurde jedem sofort verständlich und war selbst erklärend. Mit diesen Eigenschaften konnte Trello immer einen guten Überblick über den aktuellen Stand liefern und erfüllte die von uns gestellten Anforderungen sehr gut.

GitHub dagegen lief nicht ganz so unproblematisch. Die Funktionsweise über die

GUI war zwar sehr logisch und einleuchtend, allerdings kam es regelmäßig zu teils merkwürdigen Fehlern. Das größte Problem entstand, wenn zwei Mitglieder die gleiche Datei bearbeiteten. Das Mergen der Datei über die GUI war nie möglich und manuelles Mergen über die Konsole funktionierte nur selten. Um dieses Problem schnell und einfach zu Lösen, hat einer der beiden seine Änderungen separat gespeichert und das Repository neu heruntergeladen. Diese Fehler führte oft zu Frustrationen. Der Grund für diese Fehler lag sowohl an unsere Unerfahrenheit mit Codeverwaltung und an der GitHub GUI. Ein weiterer frustrierender Fehler war das Überschreiben von einigen Dateien mit älterem Inhalt. All diese Probleme führten zu Überlegungen GitHub durch ein anderes Tool, wie Dropbox, zu ersetzen. Diese Überlegungen wurden jedoch wieder verworfen, da damit vermutlich noch weitere Probleme entstanden wären. Im Nachhinein wäre es besser gewesen die Git-Bash zu verwenden und eine ausführliches Tutorial zu Git zu machen.

7 Planung und Vorgehen

Bei der Planung und Entwicklung von Avalon soll Effektivität und Zielstrebigkeit eine wichtige Rolle spielen. Deshalb war es uns von Beginn an wichtig, genügend Zeit in die Planung zu investieren. Während der ersten Planungsphase haben wir entschieden, das Projektmanagement softwareseitig zu unterstützen um die Organisation und die Zusammenarbeit untereinander zu vereinfachen sowie den zeitlichen Aufwand für die Organisation zu minimieren. Außerdem haben wir entschieden Iterationen bei der Entwicklung durchzuführen, da damit Fehler leichter aufgedeckt werden und Prioritäten für wichtige Aspekte gesetzt werden können.

7.1 Projektmanagement

7.1.1 Anforderungen

Da es sich bei Avalon um ein sehr komplexes Projekt handelt, soll Zur Unterstützung des Entwicklungsprozesses haben wir uns entschieden Programme einzusetzen, die Aufgaben des Projektmanagements abnehmen bzw. erleichtern. Für uns ergaben sich zwei zentrale Anforderungen an diese Programme. Die erste Anforderung ist es einen Gesamtüberblick über den aktuellen Stand des Projektes darzustellen. Dazu gehört, was noch erledigt werden muss, was gerade gemacht wird und was bereits erledigt wurde. Die zweite Anforderung ist eine zentrale Codeverwaltung mit Versionsverwaltung, um das gemeinsame und zeitgleiche Programmierung zu ermöglichen. Außerdem soll durch Versionsverwaltung gegen Datenverluste vorgebeugt werden.

Da wir oft von unterschiedlichen Orten und Rechnern arbeiten, müssen die genannten Anforderungen immer und überall erfüllt werden. Um diese, weitere Anforderung zu erfüllen lag es nahe, die Daten in der Cloud zu speichern. Aufgrund von persönlichen Erfahrungen und Internet Recherchen haben wir

entschieden, dass wir für den Gesamtüberblick *Trello* benutzen und für die Codeverwaltung *GitHub* benutzen. Beide Tools erfüllen die oben genannten Anforderungen.

7.1.2 Einrichtung von Trello und GitHub

Die Einrichtung von Trello ist sehr einfach, da keine Software installiert werden muss und alles im Browser läuft. Zur Einrichtung ist nur eine Registrierung und die Erstellung eines gemeinsamen *Boards* notwendig. Ein solches Board besteht aus beliebig vielen Spalten, in denen die konkreten Aufgaben stehen. Unser Avalon Board hat folgenden Aufbau:

To-Do	Doing	Done
Kaufalg.	Use-Cases	Marktanalyse
...

Auch die Einrichtung von GitHub ist recht einfach. Auch hier müssen sich die Mitglieder auf der GitHub Homepage registrieren. Danach erstellt ein Mitglied ein neues Repository und fügt die anderen Mitarbeiter als Collaborators hinzu. Zur Nutzung ist es noch notwendig *Git* oder die *GitHub-GUI* auf den einzelnen Rechnern zu installieren. Dieses Tool benötigt man um das Repository herunterzuladen und Änderungen hochzuladen. Da in unserem Team nur geringe Erfahrungen mit Git vorhanden waren, entschieden wir uns die GitHub-GUI zu nutzen.

7.1.3 Praktische Erfahrungen mit Trello und GitHub

Während Entwicklung von Avalon stellte sich besonders Trello als sehr hilfreiches, einfaches und fehlerfreies Tool dar. Die Funktionsweise wurde jedem sofort verständlich und war selbst erklärend. Mit diesen Eigenschaften konnte Trello immer einen guten Überblick über den aktuellen Stand liefern und erfüllte die von uns gestellten Anforderungen sehr gut.

GitHub dagegen lief nicht ganz so unproblematisch. Die Funktionsweise über die

GUI war zwar sehr logisch und einleuchtend, allerdings kam es regelmäßig zu teils merkwürdigen Fehlern. Das größte Problem entstand, wenn zwei Mitglieder die gleiche Datei bearbeiteten. Das Mergen der Datei über die GUI war nie möglich und manuelles Mergen über die Konsole funktionierte nur selten. Um dieses Problem schnell und einfach zu Lösen, hat einer der beiden seine Änderungen separat gespeichert und das Repository neu heruntergeladen. Diese Fehler führte oft zu Frustrationen. Der Grund für diese Fehler lag sowohl an unsere Unerfahrenheit mit Codeverwaltung und an der GitHub GUI. Ein weiterer frustrierender Fehler war das Überschreiben von einigen Dateien mit älterem Inhalt. All diese Probleme führten zu Überlegungen GitHub durch ein anderes Tool, wie Dropbox, zu ersetzen. Diese Überlegungen wurden jedoch wieder verworfen, da damit vermutlich noch weitere Probleme entstanden wären. Im Nachhinein wäre es besser gewesen die Git-Bash zu verwenden und eine ausführliches Tutorial zu Git zu machen.

8 Planung und Vorgehen

Bei der Planung und Entwicklung von Avalon soll Effektivität und Zielstrebigkeit eine wichtige Rolle spielen. Deshalb war es uns von Beginn an wichtig, genügend Zeit in die Planung zu investieren. Während der ersten Planungsphase haben wir entschieden, das Projektmanagement softwareseitig zu unterstützen um die Organisation und die Zusammenarbeit untereinander zu vereinfachen sowie den zeitlichen Aufwand für die Organisation zu minimieren. Außerdem haben wir entschieden Iterationen bei der Entwicklung durchzuführen, da damit Fehler leichter aufgedeckt werden und Prioritäten für wichtige Aspekte gesetzt werden können.

8.1 Projektmanagement

8.1.1 Anforderungen

Da es sich bei Avalon um ein sehr komplexes Projekt handelt, soll Zur Unterstützung des Entwicklungsprozesses haben wir uns entschieden Programme einzusetzen, die Aufgaben des Projektmanagements abnehmen bzw. erleichtern. Für uns ergaben sich zwei zentrale Anforderungen an diese Programme. Die erste Anforderung ist es einen Gesamtüberblick über den aktuellen Stand des Projektes darzustellen. Dazu gehört, was noch erledigt werden muss, was gerade gemacht wird und was bereits erledigt wurde. Die zweite Anforderung ist eine zentrale Codeverwaltung mit Versionsverwaltung, um das gemeinsame und zeitgleiche Programmierung zu ermöglichen. Außerdem soll durch Versionsverwaltung gegen Datenverluste vorgebeugt werden.

Da wir oft von unterschiedlichen Orten und Rechnern arbeiten, müssen die genannten Anforderungen immer und überall erfüllt werden. Um diese, weitere Anforderung zu erfüllen lag es nahe, die Daten in der Cloud zu speichern. Aufgrund von persönlichen Erfahrungen und Internet Recherchen haben wir

entschieden, dass wir für den Gesamtüberblick *Trello* benutzen und für die Codeverwaltung *GitHub* benutzen. Beide Tools erfüllen die oben genannten Anforderungen.

8.1.2 Einrichtung von Trello und GitHub

Die Einrichtung von Trello ist sehr einfach, da keine Software installiert werden muss und alles im Browser läuft. Zur Einrichtung ist nur eine Registrierung und die Erstellung eines gemeinsamen *Boards* notwendig. Ein solches Board besteht aus beliebig vielen Spalten, in denen die konkreten Aufgaben stehen. Unser Avalon Board hat folgenden Aufbau:

To-Do	Doing	Done
Kaufalg.	Use-Cases	Marktanalyse
...

Auch die Einrichtung von GitHub ist recht einfach. Auch hier müssen sich die Mitglieder auf der GitHub Homepage registrieren. Danach erstellt ein Mitglied ein neues Repository und fügt die anderen Mitarbeiter als Collaborators hinzu. Zur Nutzung ist es noch notwendig *Git* oder die *GitHub-GUI* auf den einzelnen Rechnern zu installieren. Dieses Tool benötigt man um das Repository herunterzuladen und Änderungen hochzuladen. Da in unserem Team nur geringe Erfahrungen mit Git vorhanden waren, entschieden wir uns die GitHub-GUI zu nutzen.

8.1.3 Praktische Erfahrungen mit Trello und GitHub

Während Entwicklung von Avalon stellte sich besonders Trello als sehr hilfreiches, einfaches und fehlerfreies Tool dar. Die Funktionsweise wurde jedem sofort verständlich und war selbst erklärend. Mit diesen Eigenschaften konnte Trello immer einen guten Überblick über den aktuellen Stand liefern und erfüllte die von uns gestellten Anforderungen sehr gut.

GitHub dagegen lief nicht ganz so unproblematisch. Die Funktionsweise über die

GUI war zwar sehr logisch und einleuchtend, allerdings kam es regelmäßig zu teils merkwürdigen Fehlern. Das größte Problem entstand, wenn zwei Mitglieder die gleiche Datei bearbeiteten. Das Mergen der Datei über die GUI war nie möglich und manuelles Mergen über die Konsole funktionierte nur selten. Um dieses Problem schnell und einfach zu Lösen, hat einer der beiden seine Änderungen separat gespeichert und das Repository neu heruntergeladen. Diese Fehler führte oft zu Frustrationen. Der Grund für diese Fehler lag sowohl an unsere Unerfahrenheit mit Codeverwaltung und an der GitHub GUI. Ein weiterer frustrierender Fehler war das Überschreiben von einigen Dateien mit älterem Inhalt. All diese Probleme führten zu Überlegungen GitHub durch ein anderes Tool, wie Dropbox, zu ersetzen. Diese Überlegungen wurden jedoch wieder verworfen, da damit vermutlich noch weitere Probleme entstanden wären. Im Nachhinein wäre es besser gewesen die Git-Bash zu verwenden und eine ausführliches Tutorial zu Git zu machen.

Literaturverzeichnis

Fowler, M. (2006), 'Continuous integration', <http://www.martinfowler.com/articles/continuousIntegration.html>. Abgerufen am 18.10.2013.

Hicks, J. (2012), 'Germany's carpooling.com proves rideshare works', <http://www.forbes.com/sites/jenniferhicks/2012/06/08/germanys-carpooling-com-proves-rideshare-works/>. Abgerufen am 18.10.2013.

Humble, J. und Farley, D. (2011), *Continuous delivery: reliable software releases through build, test, and deployment automation*, Addison-Wesley, Upper Saddle River, NJ [u.a.].

Kniberg, H. und Skarin, M. (2010), *Kanban and Scrum - Making the Most of Both*, Lulu Enterprises Incorporated.

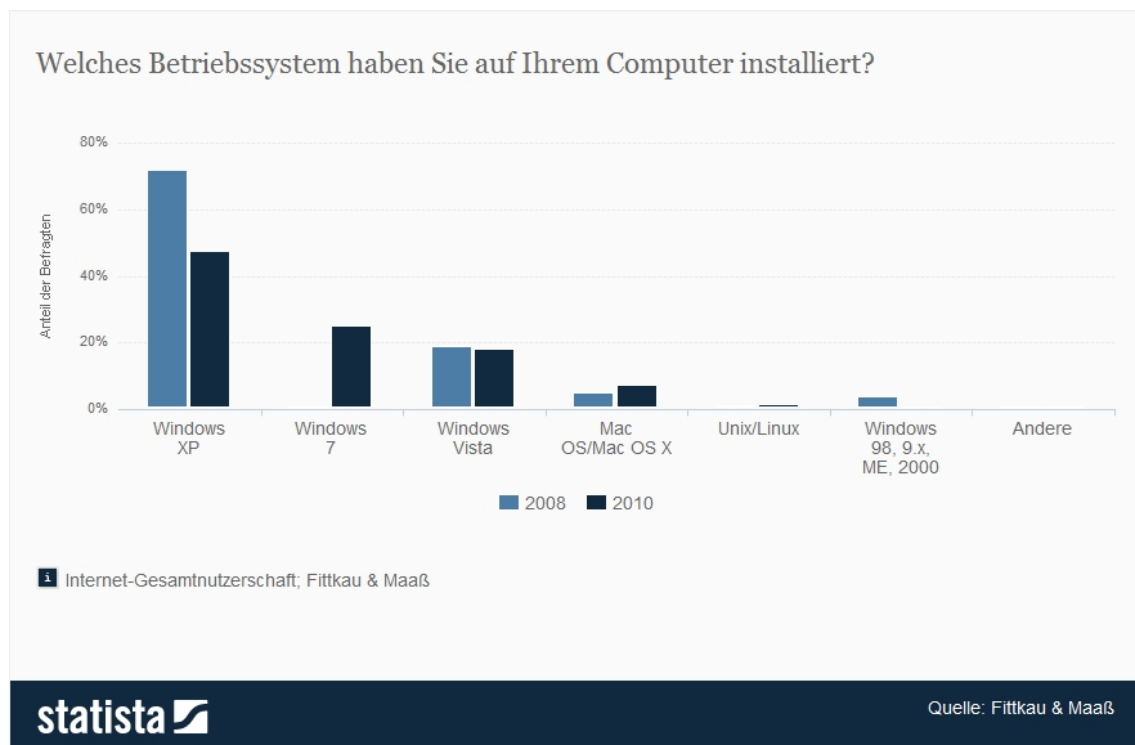
SAP AG (2013), 'SAP Geschäftsbericht 2012', <http://www.sap.com/corporate-de/investors/pdf/SAP-2012-Geschaeftsbericht.pdf>. Abgerufen am 18.10.2013.

Swartout, P. (2012), *Continuous Delivery and DevOps a Quickstart guide.*, Packt Pub., Birmingham.

Wiest, S. G. (2011), *Continuous Integration mit Hudson: Grundlagen und Praxiswissen für Einsteiger und Umsteiger*, dpunkt-Verl., Heidelberg.

Anhang

Abschnitt A Umfrage:



Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Projektarbeit 1 mit dem Thema *Optimierung des Release-Prozesses bei SAP TwoGo mithilfe von Continuous Delivery* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Projektarbeit 1 bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, 11. November 2013

Johannes Haaß