



Software Engineering Übung 6

Sprint 2 und Aufgabenbeschreibung für Sprint 3

1 Informationen

1.1 Daten

- Ausgabe Di 19.11.2013
- Abgabe So 1.12.2013 bis 23:59 Uhr
- Besprechung am Di 3.12.2013 um 12:15 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex [N] [TeamName].pdf** abgegeben werden, wobei [N] die Nummer der Übung ist und [TeamName] der Name des Teams ist. Die PDF Datei sollte ausserdem ebenfalls Ihren Namen und Matrikelnummer beinhalten.

Bitte senden Sie Ihre Lösungen via OLAT. Falls Sie zusätzliche Abgabematerialien haben, senden Sie diese als Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen Sie das Archiv anhand der oben erwähnten Konventionen.

Alle Teilaufgaben sind als Gruppenarbeit in Ihrer Übungsgruppe zu bearbeiten. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Überblick und Ziele

In Übung 5 haben Sie die Sprint-Auftragsliste (*Sprint Backlog*) für Sprint 2 erstellt, inklusive der zu implementierenden Benutzergeschichten (*User Stories*) und deren Prioritäten. Dann haben Sie die Benutzergeschichten für Sprint 2 auf einzelne Aufgaben bzw. Arbeitseinheiten (*Tasks*) heruntergebrochen und diese den einzelnen Gruppenmitgliedern zugeteilt.

Diese Übung besteht nun aus dem Ausführen der Aufgabenbeschreibung für Sprint 2, und resultiert sowohl in einer lauffähigen, getesteten und bereitgestellten (deployed) Applikation, als auch in der Aufgabenbeschreibung für Sprint 3.

Deshalb werden Sie die meiste Zeit dieser Übung für das Implementieren der von Ihnen beschriebenen Aufgaben und für das Testen verwenden. Nachdem die Implementierung gegen Ende des Sprints vollendet ist, werden Sie den Sprint 3 planen und die Aufgabenverteilung (*Work breakdown*) niederschreiben müssen. Es wird dringend empfohlen, dass Sie dies erst am Ende des Sprint 2 machen, damit Sie nicht behobene Fehler, ungelöste Arbeitseinheiten und andere in Sprint 2 aufgetretenen Probleme in der Planung von Sprint 3 berücksichtigen können.

3 Aufgabenstellung

Alle Teilaufgaben basieren auf der Anforderungsspezifikation einer Applikation zur Visualisierung von Abstimmungen, welche Sie in Übung 2 erstellt haben, sowie auf den zugehörigen Informationen in der Aufgabenstellung von Übung 2, und der Besprechung mit dem Auftraggeber am 8.10. Wenn Sie Unklarheiten haben, dann benutzen Sie das dazu eingerichtete Online Forum in OLAT oder wenden sich per E-Mail an ihren Auftraggeber.

Wie schon früher erwähnt, verwenden wir in unserem Übungsprojekt einen hybriden Entwicklungsprozess. Die vergleichsweise ausführliche Dokumentation von Anforderungen und Architektur am Beginn der Entwicklung ist eher typisch für ein ergebnisorientiertes Phasenmodell. Die Realisierung in Inkrementen fester Dauer dagegen entspricht einer agilen Vorgehensweise und lehnt sich an Scrum an. Wir verwenden jedoch wesentliche Elemente von Scrum (z.B. Scrum-Master, Daily Scrum) nicht, weil diese sich im gegebenen Kontext (viele Übungsgruppen, begrenzte Arbeitskapazitäten von Teams und Betreuern) nicht sinnvoll realisieren lassen.

Wir haben diese hybride Vorgehensweise bewusst gewählt, um den im Rahmen eines Übungsprojekts erzielbaren Lernerfolg zu optimieren.

Teilaufgabe 6.1: Das Ziel von Sprint 2 (1 Punkt)

Basierend auf den bisher erhobenen Anforderungen und dem vom Produkteigner (gespielt von den Kursassistenten) erhaltenen Input am 19. November soll Ihr Team gemeinsam das Ziel des Sprints festlegen. Beschreiben Sie kurz, was der Sprint zu erreichen versucht. Beispiel: „Implementiert grundlegende Einkaufswagen-Funktionalität mit dem Hinzufügen, Entfernen und Aktualisieren von Stückzahlen.“ Das Sprint-Ziel kann verwendet werden, um Aussenstehende kurz und prägnant über den Sprint zu berichten. Es gibt ständig Interesseneigner (*Stakeholders*), die wissen wollen, woran das Team gerade arbeitet, aber keine Details über die einzelnen Benutzergeschichten in der Produkt-Auftragsliste wissen müssen. Der Erfolg des Sprints wird später während der Sprint Review Sitzung anhand des Sprint-Ziels beurteilt (3.12.2013), und nicht anhand der einzelnen Benutzergeschichten.

Die Abgabe dieser Teilaufgabe besteht aus einem Satz, welcher das Ziel des Sprints beschreibt.

Teilaufgabe 6.2: Durchführung von Sprint 2 (8 Punkte)

Das Hauptziel dieser Aufgabe ist, am Ende von Sprint 2 ein funktionierendes, mit der Google App Engine bereitgestelltes (deployed) Pilotsystem zu haben. Der Fokus von Sprint 2 soll auf den für das Funktionieren Ihres Systems kritischen Benutzergeschichten liegen – einerseits denjenigen, welche Sie für Sprint 1 geplant, aber nicht oder nicht ganz geschafft haben, und andererseits den wichtigsten noch nicht bearbeiteten Benutzergeschichten. Nach der Planung in Übung 5 sollte jedes Teammitglied ungefähr dieselbe Auslastung haben. Falls Sie nach dem Erhalt des Feedbacks am 19. November das Gefühl haben, die Aufgaben neu verteilen oder anpassen zu müssen, dürfen Sie dies tun. Vergessen Sie nicht, die Änderungen zu dokumentieren und bei der Abgabe dieser Übung ebenfalls mitzuliefern.

Ihre Schätzungen sind wahrscheinlich nicht perfekt, aber Sie können Benutzergeschichten und Arbeitseinheiten später im Team umverteilen. Es ist nicht ausgeschlossen, dass die Implementierung einzelner Benutzergeschichten sich als aufwändiger herausstellt als Sie geschätzt haben, was zur Folge hat, dass Sie nicht alle der für diesen Sprint ausgewählten Benutzergeschichten implementieren können. Stellen Sie in diesem Fall sicher, dass Sie am Ende des Sprints alle kritischen und wichtigsten Benutzergeschichten implementieren.

Teilen Sie die Arbeit entsprechend Ihrer aktualisierten Aufgabenverteilungsliste auf die Teammitglieder auf. Führen Sie die Ihnen zugewiesenen Aufgaben aus, indem Sie entsprechende Software entwerfen, codieren und gemäss Teilaufgabe 6.3 testen.

Integrieren Sie die getesteten Module mit dem Google Web Toolkit als Rahmen und erzeugen Sie ein Pilotsystem, welches in Ihrem Browser als Applikation funktionsfähig ist. Zusätzlich müssen Sie die Applikation für diesen Sprint mit Google App Engine bereitstellen (deploy). Die folgenden zwei Tutorien helfen Ihnen besser zu verstehen wie die ganze Web Architektur des Systems aufgebaut ist, wie die verschiedenen Komponenten miteinander kommunizieren und wie die Daten gespeichert werden.

Das erste Tutorial erklärt den Remote Procedure Call (RPC) und die Kommunikation zwischen Client und Server. Befolgen Sie die unter <http://code.google.com/webtoolkit/doc/latest/tutorial/RPC.html> beschriebenen Schritte, und vervollständigen Sie die StockWatcher Applikation, die Sie in Übung 2 angefangen haben. Dabei sollten Sie die Applikation so implementieren, dass die Werte vom Server aktualisiert werden – der RPC sollte den Code auf der Serverseite aufrufen. Um einen Überblick über die Client-Server Kommunikation zu erhalten, empfehlen wir Ihnen ebenfalls den folgenden Überblick <https://developers.google.com/web-toolkit/doc/2.4/tutorial/clientserver> durchzulesen.

Das zweite Tutorial (<https://developers.google.com/web-toolkit/doc/2.4/tutorial/appengine>) führt Sie durch die notwendigen Schritte für das Bereitstellen (deployment) mit der Google App Engine. Dabei werden Sie die Authentifizierungsmechanismen von Google verwenden.

Die Abgabe dieser Teilaufgabe besteht aus einem funktionierenden, gemäss Teilaufgabe 6.3 getesteten und mit der Google App Engine bereitgestellten Pilotsystem Ihrer geplanten Applikation, welche Ihr Team dem Produkteigner (Kursassistenten) am 3. Dezember präsentiert.

Hinweise:

- Die Arbeiten der Teilaufgaben 6.2, 6.3a und 6.3b sollten *zeitlich verzahnt* und *nicht* sequenziell nacheinander ausgeführt werden.
- Planen Sie die Implementierung so, dass in Ihrem Team Arbeitskapazität für den Systemtest sowie die Suche und Behebung von Defekten (Teilaufgabe 6.3 c) verbleibt.
- Ihr Team wird *nicht* mit Punktabzügen bestraft, wenn Sie in dieser Phase Benutzergeschichten oder Arbeitseinheiten auf Sprint 3 verschieben müssen. Falls Sie jedoch kein lauffähiges, getestetes Pilotsystem vorweisen können, ziehen wir Ihnen Punkte ab.

Teilaufgabe 6.3: Testen (7 Punkte)

a) Komponententest (3 Punkte)

Schreiben Sie einen Satz von JUnit Tests, welche alle Methoden überdecken, die Sie in diesem Sprint neu geschrieben haben, d.h. mindestens ein Test pro Methode. Falls einige der Methoden Ausnahmen (exceptions) werfen, decken Sie auch diese mit Tests ab. Wir empfehlen Ihnen, den Code (vgl. Teilaufgabe 6.2) und die zugehörigen JUnit Tests parallel zu entwickeln. Die Entwicklung einer Methode oder Klasse ist erst abgeschlossen, wenn die zugehörigen JUnit Tests keine Fehler anzeigen. Falls Sie Software ändern, die Sie in Sprint 1 erstellt haben, müssen Sie die zugehörigen, in Sprint 1 erstellten JUnit Tests anpassen. Führen Sie nicht nur die neuen und geänderten JUnit Tests aus, sondern *sämtliche* bestehenden JUnit Tests. Mit diesem Regressionstest prüfen Sie, ob die Erweiterung Ihres Systems mit neuer Software möglicherweise Fehler in der bestehenden Software verursacht.

b) Testvorschrift für den Systemtest (2 Punkte)

Erweitern Sie die Testvorschrift, die Sie in Übung 5 für den manuellen Systemtest von Sprint 1 erstellt haben, um Testfälle für die in Sprint 2 implementierte Software. Erneut geht es primär darum, mit diesem Test das Verhalten Ihres Systems und die Benutzerschnittstelle zu testen. Orientieren Sie sich bei der Auswahl der Testfälle an den Abnahmekriterien, die Sie zu den für Sprint 2 ausgewählten Benutzergeschichten geschrieben haben. Falls Sie Software ändern, die Sie in Sprint 1 erstellt haben, müssen Sie die zugehörigen Systemtestfälle anpassen.

c) Durchführung des Systemtests (2 Punkte)

Führen Sie den Systemtest mit der in Teilaufgabe 6.3b erstellten Testvorschrift durch und dokumentieren Sie das Resultat. Es sind *alle* Testfälle, d.h. die bisherigen Testfälle aus Sprint 1 und die in

Teilaufgabe 6.3b neu erstellten bzw. geänderten Testfälle zu testen. Wenn der Test Fehler findet, suchen Sie die fehlerverursachenden Defekte und beheben diese. Danach ist der Systemtest zu wiederholen. Wenn Ihnen im Rahmen der für den Sprint verfügbaren Arbeitskapazität keine Zeit mehr für die Defektsuche und -behebung bleibt, erstellen Sie für jeden nicht behobenen Fehler einen Fehlerbericht (bug report). Dieser besteht aus folgenden Feldern: Fehler-ID, Kurzbeschreibung des Fehlers in wenigen Worten, Nummer des Testfalls, bei dem der Fehler auftritt, Verfasser, Priorität und Aufwand. Die Felder für Priorität und Aufwand bleiben vorerst leer.

Die Abgabe von Teilaufgabe 6.3 besteht aus den von Ihnen geschriebenen JUnit Tests, der Systemtestvorschrift mit den eingetragenen Befunden der letzten Durchführung des Systemtests sowie gegebenenfalls den Fehlerberichten für alle am Ende des Sprints nicht behobenen Fehler.

Teilaufgabe 6.4: Planungsspiel Sprint 3 (Sprint Planning Game) (2 Punkte)

Nachdem Sie die Implementierung (d.h. Codierung, Integration, Test und Bereitstellen (deployment)) für Sprint 2 beendet haben, ist es jetzt an der Zeit, Sprint 3 zu planen. Zuerst sollten Sie vergleichen, was Sie anfänglich für den Sprint 2 geplant haben, und was Sie davon während des Sprints erreicht haben. Wie in Teilaufgabe 6.2 erwähnt, haben Sie möglicherweise nicht alle Benutzergeschichten oder Arbeitseinheiten vollenden können. Und überdies haben Sie möglicherweise Fehler identifiziert. Diese Dinge sollten Sie analysieren und als Ergebnis dieser Analyse gegebenenfalls die bestehenden Aufwandschätzungen für die noch nicht implementierten bzw. die in Sprint 2 nicht geschafften Benutzergeschichten anpassen. Dann erstellen Sie die Sprint-Auftragsliste (*Sprint Backlog*) für den Sprint 3. Hierzu führen Sie folgende Tätigkeiten durch:

- Schätzen Sie den Aufwand für die geplanten, aber in Sprint 2 nicht realisierten Benutzergeschichten und Arbeitseinheiten neu ab. Überprüfen Sie auch, aufgrund der in Sprint 2 gemachten Erfahrungen, die Aufwandschätzungen für die übrigen noch nicht implementierten Benutzergeschichten und nehmen Sie gegebenenfalls Anpassungen vor.
- Überprüfen Sie die Prioritäten aller noch nicht implementierten Benutzergeschichten und nehmen Sie gegebenenfalls Anpassungen vor (in einem realen Projekt wäre dies durch den Produkteigner zu machen).
- Schätzen Sie für jeden Fehlerbericht den Aufwand für die Fehlerbehebung und priorisieren Sie den Fehlerbericht.
- Falls Sie während des zweiten Sprints neue Benutzergeschichten entdeckt haben, welche Sie in Sprint 3 realisieren wollen, so schreiben Sie diese Geschichten, schätzen ihren Aufwand ab und priorisieren sie.
- Wählen Sie nun diejenigen Benutzergeschichten und Fehlerberichte aus, welche Sie in Sprint 3 bearbeiten wollen, und erstellen Sie die Auftragsliste (Sprint Backlog) für Sprint 3 in tabellarischer Form (gemäss Tabelle 4.1 aus Übung 4). *Der Schwerpunkt von Sprint 3 soll auf Konsolidierung und Fehlerbehebung gelegt werden; Erweiterungen kommen erst in zweiter Priorität.*

Die Abgabe dieser Teilaufgabe besteht aus der Tabelle mit der Sprint-Auftragsliste für Sprint 3.

Teilaufgabe 6.5: Aufgabenverteilung für Sprint 3 (2 Punkte)

Ähnlich wie Sie dies in Übung 4, Teilaufgabe 4.5 für Sprint 1 vorbereitet haben, müssen Sie die ausgewählten Benutzergeschichten in einzelne Arbeitseinheiten herunterbrechen, welche Sie dann den Teammitgliedern zuordnen. Stellen Sie das Ergebnis tabellarisch dar (gemäss Tabelle 4.2 in Übung 4).

Die Abgabe dieser Teilaufgabe besteht aus der Tabelle mit der Aufgabenverteilung (*Work breakdown*) für Sprint 3. Die Durchführung von Sprint 3 wird Gegenstand der Übung 7 sein.

Ein paar Tipps: Wie Sie sich für die Diskussion am 3. Dezember vorbereiten

In der Diskussion am 3. Dezember werden Sie den Produkteigner (Kursassistenten) treffen, welche/r beurteilt wird, welche Positionen (Benutzergeschichten, Arbeitseinheiten) als abgeschlossen betrachtet werden. Sie werden exakt 7 Minuten Zeit haben, um Ihr Produkt zu demonstrieren (falls Sie überziehen, werden Sie unterbrochen/abgewürgt). Nutzen Sie diese Zeit, um Ihr Pilotsystem zu erklären und zu zeigen, welche Benutzergeschichten aus der Sprint-Auftragsliste implementiert sind. Es wird dringend empfohlen, dass Sie Ihre Demonstration im Vorfeld üben.

Der Produkteigner wird Ihnen Fragen zu Ihren Entwurfs- und Implementierungsentscheidungen stellen. Diese Fragen werden an einzelne Teammitglieder gerichtet. Sie sollten daher nicht nur den von Ihnen selbst geschriebenen Code kennen, sondern auch über das Gesamtdesign Ihres Pilotsystems Bescheid wissen.

Nach der Demonstration und den Fragen werden Sie zusammen mit der/dem Kursassistentin/Kursassistenten Ihren Fortschritt beurteilen und diesbezüglich Fragen diskutieren: was lief gut, was kann künftig besser gemacht werden, was haben Sie gelernt, welche Konsequenzen ziehen Sie? In der Praxis ist diese Besprechung bekannt als "Sprint Retrospective Meeting".

Danach haben Sie eine kurze Besprechung zur Sprint-Planung (3 Minuten) für den nächsten Sprint. Sie werden mit dem Produkteigner einerseits über die Benutzergeschichten reden, welche Sie in Sprint 2 nicht oder nur teilweise geschafft haben und nun in Sprint 3 implementieren wollen, über die Fehlerberichte aus Sprint 2, welche Sie in Sprint 3 bearbeiten wollen, sowie über neue Geschichten, welche Sie für Sprint 3 ausgewählt haben. Hinweis: in realen Projekten mischt sich der Produkteigner nicht in die Sprintplanung und die Aufgabenverteilung ein. In der Übungssituation profitieren Sie jedoch mehr, wenn wir Ihnen Feedback zu Ihrer Sprintplanung geben.