



Software Engineering Übung 4

Benutzerschnittstellenprototyp, Vorbereitung und Planung des ersten Sprints

1 Informationen

1.1 Daten

- Ausgabe Di 22.10.2013
- Abgabe Fr 01.11.2013 bis 23:59 Uhr (**Achtung:** ausnahmsweise *drei Tage früher* als normal!)
- Besprechung am Di 05.11.2013 um 12:15 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen Ex[n] [NameA Matrikelnummer].pdf abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA Matrikelnummer] der Name und Matrikelnummer des Studenten sind. Die PDF Datei sollte außerdem ebenfalls Ihren Namen und Matrikelnummer beinhalten.

Bitte senden Sie Ihre Lösungen via OLAT. Der Betreff der Lösung sollte mit [SE EX HS13] beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, senden Sie diese als Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen Sie das Archiv anhand der oben erwähnten Konventionen.

Die Übungen sollen in Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Überblick und Ziele

In dieser Übung setzen Sie die Entwurfsarbeit in Ihrem Übungsprojekt fort, indem Sie einen Prototyp der Benutzerschnittstelle bauen. Parallel dazu bereiten Sie die Realisierung vor, indem Sie die Anforderungen priorisieren und verfeinern. Schliesslich planen Sie das erste Inkrement (Sprint) der Realisierung Ihres Projekts.

Alle Teilaufgaben sind als Gruppenarbeit in Ihrer Übungsgruppe zu bearbeiten.

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 den 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 in der Vorlesung 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. Produkteigner, 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 4.1: Priorisierung der Anforderungen (2 Punkte)

Priorisieren Sie die Einzelanforderungen in Ihrer Anforderungsspezifikation, welche Sie in Übung 2 erstellt haben. Verwenden Sie dabei eine dreistufige Skala: kritisch, wichtig, nebensächlich. Wählen Sie die Stufe „kritisch“ für eine Anforderung, wenn Ihr System ohne die Realisierung dieser Anforderung unbrauchbar wäre. Eine Anforderung, deren Nicht-Realisierung die späteren Systembenutzer erheblich einschränkt oder behindert, aber das System nicht unbrauchbar macht, priorisieren Sie auf der Stufe „wichtig“. Alle übrigen Anforderungen sind auf der Stufe „nebensächlich“.

Hinweis: In einem realen Projekt erfolgt die Priorisierung von Anforderungen stets durch die Interesseneigner. In unserem Übungsprojekt ist dies aus logistischen Gründen nicht machbar. Ihr Team muss sich daher für diese Teilaufgabe in die Rolle der Interesseneigner versetzen. Lassen Sie sich bei der Beurteilung der Wichtigkeit der einzelnen Anforderungen von folgenden Informationen leiten:

- Vision des Auftraggebers (siehe Übung 2, Kapitel 3 der Aufgabenstellung)
- Antworten der Umfrage bei potenziellen Interessenten (siehe Übung 2, Kapitel 3 der Aufgabenstellung)
- Antworten der Auftraggeber auf Ihre Fragen in der Übungsstunde vom 8. Oktober 2013.

Teilaufgabe 4.2: Erstellung von Benutzergeschichten (*User Stories*) (5 Punkte)

Verfeinern Sie die Einzelanforderungen aus Ihrer Anforderungsspezifikation durch die Erstellung von Benutzergeschichten (user stories). Sie müssen in dieser Teilaufgabe mindestens 15 Geschichten erstellen.

Jede Benutzergeschichte besteht aus drei Elementen: (1) einem Satz nach dem Schema

Als <Rolle> will ich <meine Anforderung> [so dass <Nutzen für>]

oder auf Englisch

As a <role> I want <my requirement> [so that <benefit>],

(2) einer Reihe von Abnahmekriterien, welche erfüllt sein müssen, damit die Benutzergeschichte als adäquat implementiert akzeptiert wird, und (3) folgenden Metadaten: Id-Nr (eindeutige Identifikation), Autor, Erstellungsdatum, Id-Nr der verfeinerten Anforderung, Priorität, Aufwand. Fügen Sie Ihre Benutzergeschichten in das bisher leere Kapitel 6.2 Ihrer Anforderungsspezifikation ein. Ein Beispiel einer Benutzergeschichte finden Sie unten.

Hinweise:

- Sie erstellen in dieser Aufgabe nur eine Teilmenge aller Benutzergeschichten. In den kommenden Übungen werden Sie weitere Benutzergeschichten schreiben.
- Benutzergeschichten sind nicht in Stein gemeißelt: nach jedem Sprint können Sie bestehende Benutzergeschichten, welche noch nicht implementiert sind, ändern, ergänzen oder verfeinern.

- Denken Sie daran, dass Sie auch Benutzergeschichten für die Codierung der Benutzerschnittstelle brauchen.
- Konzentrieren Sie sich vor allem auf diejenigen Benutzergeschichten, die für das Zustandekommen einer ersten lauffähigen Systemversion erforderlich sind und die Sie voraussichtlich im ersten Sprint implementieren können.
- Die Metadatenfelder „Priorität“ und „Aufwand“ bleiben zunächst leer; sie werden in Teilaufgabe 4.4 ausgefüllt.

Beispiel einer Benutzergeschichte

Für das in Folienkapitel 4.1 der Vorlesung vorgestellte System zur Gewährleistung der Zugangskontrolle zu den Transportanlagen eines Skigebiets wurde die in Bild 4.1 dargestellte Benutzergeschichte erstellt.

Story ID: S-18	Refers to requirement: R 4.2
Story: As a skier, I want to pass the chairlift gate so that I get access without presenting, scanning or inserting a ticket at the gate.	
Acceptance criteria: <ul style="list-style-type: none"> • Recognizes cards worn anywhere in a pocket on the left side of the body in the range of 50 cm to 150 cm above ground. • If card is valid: unlocks turnstile and flashes a green light for five seconds or until the turnstile is moved. • If card is invalid: doesn't unlock gate and flashes a red light for five seconds. • Time from card entering the sensor range until unlock and flash red or green is less than 1.5 s (avg) & 3 s (max). • The same card is not accepted twice within an interval of 180 s. 	
Author: Dan Downhill	Date: 2013-09-20
Priority: Critical	Effort: not yet assigned

Bild 4.1. Beispiel einer Benutzergeschichte

Teilaufgabe 4.3: Benutzerschnittstellen-Prototyp (5 Punkte)

Erstellen Sie einen Prototyp der Benutzerschnittstelle Ihrer Applikation. Sie sollen in dieser Teilaufgabe keinen Code schreiben, sondern mit einfachen Mitteln (z.B. html-Seiten, Powerpoint Folien oder abfotografierte Skizzen mit durch Pfeile symbolisierter Navigation) einen Wegwerf-Prototypen bauen. Dieser Prototyp dient später als Vorgabe für die Implementierung der Benutzerschnittstelle.

Teilaufgabe 4.4: Planungsspiel Sprint 1 (*Sprint Planning Game*) (4 Punkte)

In dieser Teilaufgabe geht es darum, diejenigen Benutzergeschichten auszuwählen, welche sie im ersten Realisierungskrement (Sprint 1) implementieren. Hierzu muss als erstes die für einen Sprint verfügbare Arbeitskapazität abgeschätzt werden. Wir erwarten von Ihnen, dass jedes Gruppenmitglied in jeden der drei Sprints einen Implementierungsaufwand von je 12 Arbeitsstunden investiert. In einer Vierergruppe steht damit eine Arbeitskapazität von 48 Personentagen pro Sprint zur Verfügung. Damit Sie entscheiden können, welche und wie viele Benutzergeschichten in einem Sprint realisierbar sind, müssen Sie den Implementierungsaufwand (d.h. den Aufwand für Detailentwurf, Codierung, Test und Integration) für jede Benutzergeschichte abschätzen. Dies geschieht in Teilaufgabe 4.4 a. Ferner müssen Sie bei der Auswahl der Benutzergeschichten die Priorität der einzelnen Geschichten berücksichtigen. In Teilaufgabe 4.4 b muss daher, basierend auf der Priorisierung aus Teilaufgabe 4.1, eine Feinpriorisierung vorgenommen werden. In Teilaufgabe 4.4 c schliesslich erfolgt die Auswahl der in Sprint 1 zu implementierenden Benutzergeschichten.

a) Aufwandschätzung (1 Punkt)

Schätzen Sie für jede der in Teilaufgabe 4.2 erstellten Benutzergeschichten den Implementierungsaufwand (Detailentwurf, Codierung, Test und Integration) in Personentagen. Notieren Sie das Ergebnis im Metadatenfeld „Aufwand“ der jeweiligen Benutzergeschichte.

b) Feinpriorisierung (1 Punkt)

Erstellen Sie eine Feinpriorisierung der in Teilaufgabe 4.2 erstellten Benutzergeschichten. Verwenden Sie wie in Teilaufgabe 4.1 eine dreistufige Skala (kritisch, wichtig, nebensächlich). Orientieren Sie sich dabei an der Priorisierung der Anforderungen (vgl. Teilaufgabe 4.1). Notieren Sie das Ergebnis im Metadatenfeld „Priorität“ der jeweiligen Benutzergeschichte.

Hinweis: Die Anforderungsprioritäten können nicht einfach 1:1 übernommen werden. Beispielsweise kann eine als „kritisch“ priorisierte Anforderung in vier Benutzergeschichten verfeinert worden sein, von denen nur eine wirklich kritisch ist, während die anderen drei als wichtig oder gar als nebensächlich zu priorisieren sind.

c) Erstellung der Sprint-Auftragsliste (*Sprint Backlog*) (2 Punkte)

Erstellen Sie auf der Grundlage der in den Teilaufgaben 4.2, 4.4 a und 4.4 b erarbeiteten Ergebnisse die Auftragsliste (Sprint Backlog) für Sprint 1 Ihres Projekts. Dabei müssen Sie folgende Kriterien berücksichtigen:

1. Am Ende des ersten Sprints müssen Sie ein *lauffähiges* System haben. Dieses kann bezüglich Funktionsumfang und Benutzungskomfort noch sehr eingeschränkt sein, aber es muss lauffähig sein und mindestens eine Funktion für einen Endbenutzer-Interesseneigner implementieren.
2. Die Summe der geschätzten Aufwendungen für die ausgewählten Benutzergeschichten muss kleiner oder gleich der verfügbaren Arbeitskapazität (48 Personenstunden in einer Vierergruppe) sein.
3. Schöpfen Sie die Arbeitskapazität Ihres Teams aus: Beispielsweise wäre es falsch, bei einer Kapazität von 48 Personentagen Geschichten im Gesamtumfang von nur 42 Personentagen auszuwählen mit der Begründung, man habe dann noch 6 Personenstunden Reserve.
4. Allfällige Abhängigkeiten zwischen Benutzergeschichten müssen berücksichtigt werden. Wenn beispielsweise Benutzergeschichte C von der Realisierung der Benutzergeschichten A und B abhängt, dann kann C nur gewählt werden, wenn A und B ebenfalls gewählt werden oder in einem früheren Sprint bereits realisiert worden sind.
5. Um den Nutzen für die Interesseneigner zu maximieren, wählen Sie bevorzugt solche Benutzergeschichten aus, die als kritisch oder als wichtig priorisiert sind.

Stellen Sie das Ergebnis tabellarisch entsprechend Tabelle 4.1 dar.

Hinweis: Es ist möglich, dass Sie bei der Bearbeitung von Teilaufgabe 4.4 c feststellen, dass Sie in Teilaufgabe 4.2 zu wenig Benutzergeschichten geschrieben haben, oder dass Ihre Geschichten zu wenig feingranular sind. In diesem Fall müssen Sie nochmals zur Bearbeitung von Teilaufgabe 4.2 zurückkehren und weitere Benutzergeschichten schreiben bzw. zu umfangreiche Geschichten in mehrere kleinere Geschichten aufteilen.

Tabelle 4.1. Auftragsliste (Sprint Backlog)

Auftragsliste (Sprint Backlog)		Sprint:	Team:	
Id	Text der Geschichte	Priorität	Aufwand	Status

Teilaufgabe 4.5: Aufgabenverteilung für Sprint 1 (4 Punkte)

Brechen Sie jede Benutzergeschichte aus der Auftragsliste in einzelne Arbeitseinheiten, sogenannte *Tasks* herunter und legen Sie für jede Arbeitseinheit fest, welche Person in Ihrer Gruppe diese Arbeitseinheit bearbeiten wird. Stellen Sie das Ergebnis tabellarisch entsprechend Tabelle 4.2 dar. Die Summe der Aufwände für die Tasks einer Benutzergeschichte muss gleich dem für diese Geschichte geschätzten Aufwand sein.

Hinweis: In realen Projekten, welche mit einem agilen Prozess entwickelt werden, wird in der Regel keine explizite Aufgabenverteilung erstellt, sondern die Teammitglieder picken sich Ihre Aufgaben selbständig aus der Auftragsliste. Die Koordination erfolgt über eine Kurzbesprechung zu Beginn jedes Arbeitstags. In einer Übungssituation, in der sich das Team nicht täglich koordiniert, funktioniert dieses Vorgehen nicht zuverlässig. Aus diesem Grund verlangen wir, dass Sie eine explizite Aufgabenverteilung vornehmen.

Tabelle 4.2. Aufgabenverteilung (Work breakdown)

Aufgabenverteilung (Work breakdown) Sprint:		Team:			
Task Nr.	Geschichte Nr.	Taskbezeichnung	Aufwand	Person	Status

Nach Abgabe der Übung muss jedes Mitglied der Gruppe in der Lage sein den Tutoren Auskunft über alle Teile der Übung zu geben und eventuelle Fragen beantworten. Sie werden die Möglichkeit erhalten in der Gruppenbesprechung am 05.11.2013 Ihre Lösungen zu präsentieren.