

Lucerne University of
Applied Sciences and Arts

**HOCHSCHULE
LUZERN**

Modul APPE | TA.APPE.F1101

APPE Filialen-Bestellsystem

P r o j e k t p l a n

Projekt	APPE Filialen-Bestellsystem	
Dokument	Projektplan	
Schule	Hochschule Luzern, Technik & Architektur	
Modul	TA.APPE.F1101	
Projektteam	<p>Galliker Thomas Studiengang Informatik (BB) Panorama 6123 Geiss Tel. +41 79 504 80 70 thomas.galliker@stud.hslu.ch</p> <p>Stocker Elias Studiengang Informatik (BB) Schybenacherweg 12 6285 Hitzkirch Tel. +41 79 603 10 89 elias.stocker@stud.hslu.ch</p>	<p>Gasser Martin Studiengang Informatik (BB) Mettenwilhöhe 4 6275 Ballwil</p> <p>Tel. +41 79 755 04 32 martin.gasser@stud.hslu.ch</p>
Dozenten	Prof. Jörg Hofstetter	
Letzte Änderung	3. Juni 2011, 11:28:00 Uhr	

Änderungsprotokoll

Version	Datum	Autor	Beschreibung
0.1	10.03.2011	ste	Initialversion von Vorlage erstellt
0.2	11.03.2011	gat	Rollen und Zuständigkeiten verteilt, grobe Projektplanung
0.3	17.03.2011	ste	Allgemeine Überarbeitung
0.4	31.03.2011	ste	Meilensteinberichte
0.5	29.04.2011	ste	Diverse Nachtragungen / Updates
0.6	08.05.2011	gat	Meilensteinberichte nachgeführt, Controlling (Ist-Aufwände ergänzt)
0.7	02.06.2011	gat	Aufwandauswertung, Lessons Learned

Inhalt

1	Einleitung.....	5
1.1	Ziel & Zweck dieses Dokuments.....	5
1.2	Projektübersicht	5
1.3	Begriffe & Abkürzungen	5
2	Projektorganisation.....	6
2.1	Projektstruktur.....	6
2.2	Projektmitglieder	6
2.3	Rollen & Zuständigkeiten	6
3	Planung	8
3.1	Grobplanung	8
3.2	Meilensteine.....	8
3.3	Rahmenplan	9
4	Arbeitspakete und Aufwandschätzung	10
5	Meilensteinberichte	12
5.1	Meilenstein 1: SW1	12
5.2	Meilenstein 2: SW4	12
5.3	Meilenstein 3: SW7	12
5.4	Meilenstein 4: SW10	12
5.5	Meilenstein 5: SW13	12
6	Risikomanagement.....	13
7	Projektunterstützung	14
7.1	Konfigurationsmanagement.....	14
7.2	Dokumentationsplan	14
8	Projektabschluss	15
8.1	Lessons Learned.....	15
8.2	Freigabeprotokoll	16

Abbildungsverzeichnis

Abbildung 1: Projektstruktur	6
Abbildung 2: Gantt-Diagramm des Rahmenplans	9

Tabellenverzeichnis

Tabelle 1: Abkürzungserklärungen	5
Tabelle 2: Koordinaten der Projektmitglieder	6
Tabelle 3: Rollen & Zuständigkeiten	7
Tabelle 4: Detaillierte Aufschlüsselung der Meilensteine.....	8
Tabelle 5: Aufwandschätzung und Controlling	11
Tabelle 1: Risikoanalyse: Risiken und geplante Gegenmassnahmen.....	13
Tabelle 2: Liste der Dokumente, welche im Rahmen des Projekts entstanden sind.....	14

1 Einleitung

1.1 Ziel & Zweck dieses Dokuments

Dieser Projektplan ist das zentrale Dokument für das Projekt Management im APPE Projekt „Filialen-Bestellsystem“.

1.2 Projektübersicht

Das Projekt „Filialen-Bestellsystem“ befasst sich mit der Konzipierung und Umsetzung eines elektronischen Bestellsystems mit dezentralen Filialen. Mitarbeiter sollen in ihren Filialen die Möglichkeit haben, das System zu verwalten. Die konkreten Anforderungen sind im Dokument „Kundenanforderung“ erfasst.

1.3 Begriffe & Abkürzungen

Abkürzung	Erklärung
HSLU	Hochschule Luzern
APPE	"Applikationsentwicklung"; Modulbezeichnung HSLU
STASS	Steuerungsausschuss
gat	Namenskürzel für Galliker Thomas
ste	Namenskürzel für Stocker Elias
gam	Namenskürzel für Gasser Martin

Tabelle 1: Abkürzungserklärungen

2 Projektorganisation

2.1 Projektstruktur

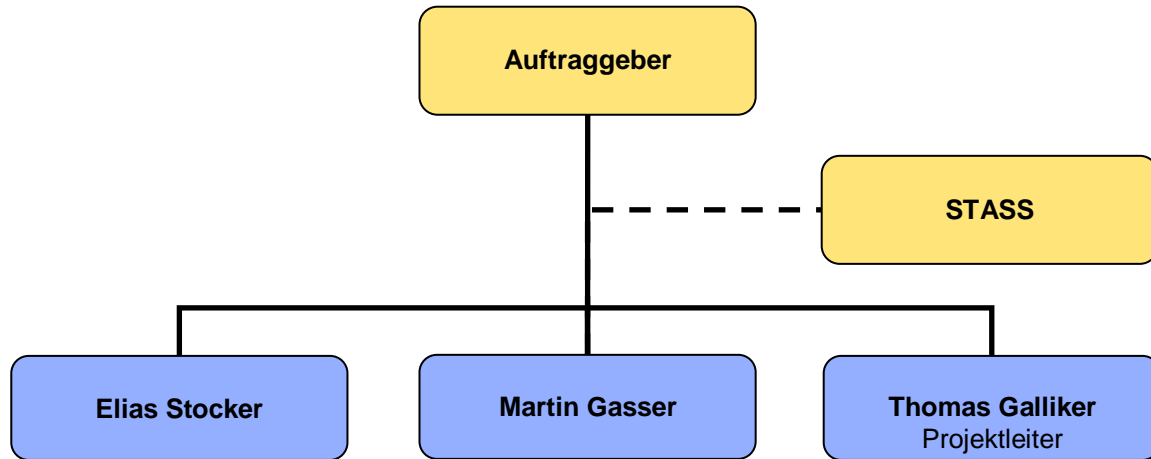


Abbildung 1: Projektstruktur

2.2 Projektmitglieder

Name / Adresse	Telefon	E-Mail
Galliker Thomas Studiengang Informatik (BB) Panorama 6123 Geiss	+41 79 504 80 70	thomas.galliker@stud.hslu.ch
Stocker Elias Studiengang Informatik (BB) Schybenacherweg 12 6285 Hitzkirch	+41 79 603 10 89	elias.stocker@stud.hslu.ch
Gasser Martin Studiengang Informatik (BB) Mettenwilhöhe 4 6275 Ballwil	+41 79 755 04 32	martin.gasser@stud.hslu.ch

Tabelle 2: Koordinaten der Projektmitglieder

2.3 Rollen & Zuständigkeiten

Rolle	Verantwortung	Beschreibung / Aufgaben
Projektspezifisch		
Projektmanagement	gat	Ist für das Projekt verantwortlich - Leitet die Sitzungen - Verteilt Aufgaben an Teammitglieder - Hält den Projektplan aktuell
Programmierung	gam	Verantwortlich für Entwicklung. Siehe genauere Aufschlüsselung weiter unten.
Testmanagement	ste	Erstellt und Ergänzt Testdokument & ist für die Tests verantwortlich - Wertet die Tests aus

Architektur	gam	Plant die SW-Architektur und erstellt das grobe Gerüst (Struktur & Files) in NetBeans - hält den Lead bei der Programmierung (genauere Definition siehe unten)
Dokumentenverwaltung	ste	Erstellt Layout aller Dokumente, hält diese aktuell (z.T. durch Input der anderen Gruppenmitglieder) - hält Struktur auf ILIAS - behält den Überblick über die Dokumente
Entwicklungsspezifisch		
Presentation Tier	ste	Die Präsentationsschicht ist verantwortlich für die Darstellung der Funktionsblöcke sowie der generierten Resultate. Konkret handelt es sich um das GUI der Anwendung.
Logic Tier	gam	Die Logikschicht implementiert die Geschäftsprozesse. Sie ist auch Vermittlungsschicht zwischen Präsentations- und Datenschicht, soll aber so wenig wie möglich von diesen beiden Schichten abhängig sein.
Data Tier	gat	Die Datenschicht stellt die Datengrundlage des Systems dar. Hier werden die einzelnen Datensätze in einem relationalen Datenbanksystem gespeichert.

Tabelle 3: Rollen & Zuständigkeiten

3 Planung

3.1 Grobplanung

Projektstart	21.02.2011 (SW1)
Projektabschluss	30.05.2011 (SW14)
Projektiterationen	2

3.2 Meilensteine

SW	Inhalt	Meilenstein
SW1	<ul style="list-style-type: none"> • Organisation der Projektgruppe • Verantwortlichkeiten und Rollen verteilen • Identifizierung und Bewertung von Risiken • Rahmenplan mit Meilensteinen 	25.02.2011 ♦
SW2	<ul style="list-style-type: none"> • Erstellung Projektmanagementplan • Grobe Arbeitsaufteilung 	
SW3	<ul style="list-style-type: none"> • Dokumentationsplan • Testplan mit Testphilosophie und wesentlichen Testaspekten • Kundenanforderungen erfassen und bewerten 	
SW4	<ul style="list-style-type: none"> • Beschreibung der wichtigsten Geschäftsabläufe (Use Cases) • Definition der Arbeitspakete • Detaillierte Aufgabenteilung • Aufwandschätzung 	18.03.2011 ♦
SW5	<ul style="list-style-type: none"> • Controlling • Softwarearchitektur definieren und dokumentieren • Systemspezifikation erstellen • ER-Modellierung • Schnittstellen zwischen Schichten definieren 	
SW6	<ul style="list-style-type: none"> • Zur Abgabe bereit: Rahmenplan 	
SW7	<ul style="list-style-type: none"> • Datenmodell erstellen • Beginn Programmierung, Erster Prototyp (vertikal) erstellt • Zur Abgabe bereit: Use Cases, Softwarearchitektur 	08.04.2011 ♦
SW8	<ul style="list-style-type: none"> • Unit Tests definieren, ausführen und protokollieren 	
SW9	<ul style="list-style-type: none"> • Entwickeln und Testen der vertikalen Prototypen 	
SW10	<ul style="list-style-type: none"> • Proof of Concept: Alle vertikalen Prototypen lauffähig • GUI Prototyp fertig • Design Entscheide verfeinern und dokumentieren 	28.04.2011 ♦
SW11	<ul style="list-style-type: none"> • System Tests durchführen und protokollieren 	
SW12	<ul style="list-style-type: none"> • Softwareverteilung dokumentieren • Betriebshandbuch erstellen 	
SW13	<ul style="list-style-type: none"> • Erstellen und Vorbereiten der Präsentation 	20.05.2011 ♦
SW14	<ul style="list-style-type: none"> • Präsentation der Projektarbeit • Projekt Deliverables abgeben 	

Tabelle 4: Detaillierte Aufschlüsselung der Meilensteine

3.3 Rahmenplan

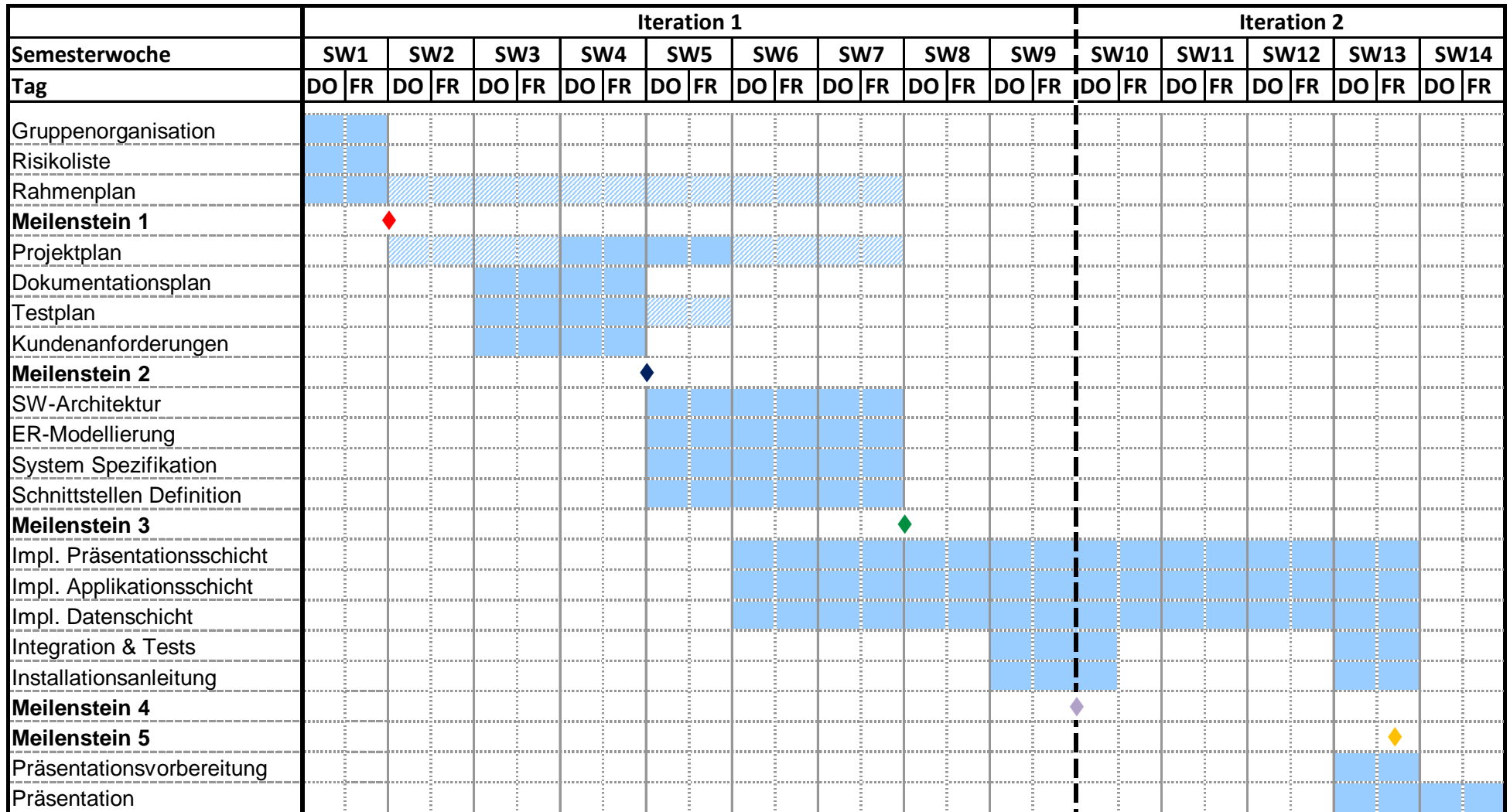


Abbildung 2: Gantt-Diagramm des Rahmenplans

4 Arbeitspakete und Aufwandschätzung

Die nachfolgenden Schätzungen beziehen sich auf den Gesamtaufwand, welcher das Team zu leisten hat. Die einzelnen Stundenangaben sind als Mannstunden zu verstehen. Um einen guten IST-SOLL Vergleich zu erlangen, wird in regelmässigen Zeitabschnitten der ungefähre IST Aufwand erfasst.

	SOLL Aufwand			IST Aufwand				Auswertung
	Schätzung 1 in [h]	Schätzung 2 in [h]	Mittelwert in [h]	SW1-3 in [h]	SW4-6 in [h]	SW7-9 in [h]	SW10-14 in [h]	Differenz in [h]
Projektmanagement								
Projekt Kick-Off	2	4	3.0	3				3.0
Risikomanagement: Identifizierung und Bewertung von Risiken; laufende Beobachtung der Risiken	4	2	3.0	2	1			3.0
Rahmenplan mit Meilensteinen und grobe Aufwandschätzung erstellen	3	3	3.0	3				3.0
Erstellen des Projektmanagementplan	8	12	10.0	5	2			7.0
Erstellen des Testplan, Definition der Testfälle	20	20	20.0		4	3		7.0
Planen und durchführen eines Reviews	6	4	5.0		1	1		2.0
Dokumentationsplan erstellen	1	1	1.0	1				1.0
Systemspezifikation erstellen und nachführen	10	12	11.0	4	2	4	4	18.0
Entwicklungsaufwand								
Software-Architektur entwickeln	12	10	11.0		6	3		9.0
Datenmodellierung	5	6	5.5		2	1		3.0
Definieren der Schnittstellen	2	3	2.5		2	2		4.0
Programmieraufwand								
- <i>Implementation Presentation Layer</i>	20	10	15.0		5	20	20	45.0
- <i>Implementation Business Logic Layer</i>	25	20	22.5		10	50	30	90.0
- <i>Implementation Data Layer & Database</i>	15	15	15.0		20	10	5	35.0

Testaufwand								
Testdaten generieren	1	1	1.0		1	2	3	5.0
Build Server integrieren, Code Coverage Messungen	6	10	8.0			3	4	7.0
Unit-Tests	20	20	20.0		5	5	15	20.0
Integrations-/Systemtests	10	10	10.0				12	12.0
Projektabschluss & Verteilung								
Betriebshandbuch erstellen	3	4	3.5				3	3.0
Software Deployment	1	1	1.0				3	3.0
Vorbereitung und Durchführung der Präsentation	7	5	6.0				5	4.0
			177.0h					224.0h

Tabelle 5: Aufwandschätzung und Controlling

5 Meilensteinberichte

5.1 Meilenstein 1: SW1

Die Gruppe ist organisiert, die Rollen sind verteilt. Erste Punkte der Risikoliste sind definiert. Bis SW2 sollte sie allerdings noch erweitert und ergänzt werden. Der Rahmenplan steht noch nicht. Durch die Zuweisung eines Verantwortlichen (gat) wird auch der Rahmenplan bis SW2 erstellt.

5.2 Meilenstein 2: SW4

Rahmenplan steht. Erste Versionen des Projektplans, der Kundenanforderungen und des Dokumentationsplan existieren. Die Erstellung des Testplans muss verschoben werden auf MS 3 (SW7). Die Umsetzung des Testplans liegt in der Verantwortung von ste.

5.3 Meilenstein 3: SW7

Unklarheiten Architektur: Wie wird der Präsentationsschicht vom Business Logic Layer getrennt? Welche Technologien können eingesetzt werden, damit der Präsentationslayer keine Kenntnisse vom Datenlayer haben muss? Wie werden die Daten zwischen den Schichten ausgetauscht? Wo sind ggf. DTO's angebracht?

Nach Rücksprache mit anderen Teams wurde klar, dass der Einsatz von DTO's zwischen Präsentationsschicht und Business Logik am meisten Sinn macht. Diese beiden Layer können mit Remoting Technologie (RMI) getrennt werden. Die Präsentationsschicht wird folglich nur ein „einfaches“ Interface kennen, hinter welcher sich die Business Logik verbirgt.

5.4 Meilenstein 4: SW10

Proof of Concept: Sind die erstellten Prototypen im System einsetzbar oder bedarf es noch weiterer Entwicklungen?

Die eingesetzte Software Architektur kann die geforderten Anforderungen erfüllen. Dank der Erstellung verschiedenster unabhängiger Prototypen konnte die technische Machbarkeit geprüft und somit die Risiken minimiert werden.

5.5 Meilenstein 5: SW13

Mit zunehmender Komplexität steigt tatsächlich die Gefahr, den Überblick zu verlieren. Speziell im Fehlerfall. So geschehen mit der generischen DTO Assemblierung. Eine Rekursion in der Umwandlung von Entity Object zu Data Transfer Object hat zu einer StackOverflowException geführt. Diese war mit Debugging nicht einfach zu finden, da die Stackgröße doch ziemlich beträchtlich ist.

Des Weiteren wurden einige neue Unit-Tests geschrieben. Der Test-First Ansatz hat sich bewährt. Auffällig war, dass die SetUp und TearDown nach jedem Unit Test ausgeführt wurden. Die Code Coverage sowie einige dringliche Warnungen konnten mit Hilfe des Buildservers behoben werden.

6 Risikomanagement

Das nachfolgende Risikomanagement beschäftigt sich mit der Identifikation von möglichen Risiken welche im Verlauf des Projekts eintreten können sowie die entsprechenden Gegenmassnahmen.

Beschreibung	Initialrisiko ¹ in SW1	Vermeidungs- aufwand ²	Gewichtung ³	Massnahmen	Risiko ¹ in SW4	Risiko ¹ in SW7	Risiko ¹ in SW10
Abweichungen im Zeitplan	3	1	3	Erst durch eine Analyse die Arbeitspakete festlegen, danach durch Synthese einen realistischen Rahmenplan erstellen. Controlling sauber nachführen.	2	1	1
Missverständnisse im Projektauftrag	1	1	1	Projektauftrag sauber lesen und alle Unklarheiten beseitigen.	1	0	0
Probleme mit dem Zugriff auf die Zentrallagerkomponente.	2	1	2	Klare, saubere und unmissverständliche Dokumentation der Schnittstellen, Kommunikation innerhalb der Projektgruppe stärken.	2	1	1
Softwarearchitektur ist zu komplex	1	2	1	Komplexität der Architektur und des Designs gering halten. Ggg. Nachweis mit Prototyp erarbeiten. Modularisierte Entwicklung.	1	0	0
Kommunikation zwischen Schichten	3	1	3	Frühzeitig Prototyp erstellen und falls nötig alternative Technologie suchen	3	2	1
Ungewollte Abhängigkeiten zwischen den Schichten	1	3	3	Architektur mit Hilfe von Komponentendiagramm aufzeichnen und mit Prototyp testen, jede Schicht in separatem Projekt entwickeln.	1	2	1
Test Cases sind ungenügend spezifiziert	3	2	1.5	Vorausschauende Spezifikation der Test Cases mit Hilfe der Use Cases.	3	1	1
Definierte Aspekte des Systemtests sind nicht durchführbar	1	1	1	Vorausschauende Definition der Testaspekte und Durchführung anhand eines Prototyps.	1	1	1

Tabelle 1: Risikoanalyse: Risiken und geplante Gegenmassnahmen

¹ Risiko (R): 1= Geringes Risiko, 2=Mittleres Risiko, 3= Hohes Risiko

² Vermeidungsaufwand (V): 1=Geringer Aufwand, 2=Mittlerer Aufwand, 3=Hoher Aufwand

³ Diejenige Massnahme, welche das grösste Gewicht erreicht, wird in erster Priorität umgesetzt.

D.h. das Gewicht stellt eine Sortierreihenfolge zur Festlegung der Umsetzungspriorität dar. Berechnung: (G) = (R) / (V)

7 Projektunterstützung

7.1 Konfigurationsmanagement

7.1.1 Versionisierung

Der gesamte Programmcode wird in NetBeans 6.9.1 erstellt und mit Hilfe eines Subversion-Plugins versionisiert. Somit können Änderungen am Code vollständig nachvollzogen und im Fehlerfall auch rückgängig gemacht werden. Der Subversion Server wird von der HSLU bereitgestellt:

<https://dev.enterpriselab.ch/education/appe.f1101.11>

Alle im Rahmen des Projekts erarbeiteten Dokumente werden von den entsprechenden Autoren manuell versionisiert. Dafür wurde eine Versionisierungstabelle am Anfang jedes Dokuments eingefügt. Um den Dokumentenaustausch zu vereinfachen und die Kommunikation für die Dokumentenkooperation zu minimieren, verwendet das Projektteam das Freeware Tool "Dropbox" (<http://www.getdropbox.com>).

7.1.2 Configuration Items

- Dokumentation
- Präsentationsschicht: fbsguiclient
- Applikationsschicht: fbsservice
- Datenschicht / Datenbank: fbsdata / fbsdb
- Betriebssystem mit Java Runtime Environment

7.2 Dokumentationsplan

Die Dokumentationen und das Projektmanagement werden anhand des Prozesses HTAgil durchgeführt und festgehalten. Die Dokumente werden mit Dropbox verwaltet. Die Versionisierung wird über die entsprechende Tabelle des jeweiligen Dokuments vorgenommen.

Dokument	Zeitraum	Kommentar zur Abgabe	Verantwortung	Autor
APPE_Projekt_Projektplan.doc	SW02-SW13	Als PDF abgeben	gam	Alle
APPE_Projekt_Systemspezifikation.doc	SW4-SW5, SW10	Als PDF abgeben	gat	Alle
APPE_Projekt_Testplan.doc	SW3, SW9	Als PDF abgeben	ste	Alle
APPE_Projekt_Testprotokoll.doc	SW4-SW10	Als PDF abgeben	ste	Alle
APPE_Projekt_Betriebshandbuch.doc	SW14	Als PDF abgeben	gat	Alle
APPE_Projekt_Präsentation.ppt	SW14	Präsentieren und als PDF abgeben	ste	Alle

Tabelle 2: Liste der Dokumente, welche im Rahmen des Projekts entstanden sind

8 Projektabschluss

8.1 Lessons Learned

Einmal mehr ist es nicht die Aufgabenstellung an sich, welche Probleme bereitete, sondern die für Java benötigten Entwicklungswerkzeuge, welche offensichtlich noch in sehr kleinen Kinderschuhen stecken. Abermals wurden Stunden um Stunden für unnötige Probleme in Netbeans verbraten. Erwähnenswert sind auch die Probleme mit dem Netbeans „WYSIWYG“ GUI Designer, welcher von uns kurzerhand als „What you see is NOT what you get“ Designer herabgestuft wurde. Die aufgewendete Zeit hätte man viel besser in die Projektarbeit stecken können.

Der Umgang mit der Java Persistence API war zu Beginn noch von Unklarheiten überdeckt. Im Laufe der Zeit wussten wir jedoch um die Tücken und Stolpersteine und konnten diese (mitunter dank der professionellen Hilfe von Dr. Olnhoff) elegant eliminieren. Erwähnenswert ist hier beispielsweise die Merge- und die Refresh-Methode, welche vom Entity Manager bereitgestellt werden: Merge sorgt dafür, dass ein Domain Object (wieder) in den Kontext des Entity Managers aufgenommen wird, während Refresh das Neueinlesen des Domain Objects von der Datenbank erzwingt. Insbesondere wenn bestimmte Attribute in Datenbanktabelle automatisch mit Trigger abgefüllt werden (bspw. SessionKey in Tabelle Session) so bietet Refresh eine gute Möglichkeit, diese generierten Werte in Domain Objects anzeigen zu lassen.

Das Konzept von Data Transfer Objects wie es von Fowler entwickelt wurde, war anfänglich unbekannt und deshalb oft hinterfragt. Der Aufwand für das Erstellen von DTO-Objekten war dem Projektteam schlicht zu gross. Später kam noch hinzu, dass für das Umwandeln von Domain Object (DO) zu Data Transfer Object (DTO) jeweils eine Assembler-Klasse hätte geschrieben werden müssen. Dank einer trickreichen Bibliothek für das generische Umwandeln von DO's zu DTO's konnten wir immerhin an dieser Stelle viele Code Zeilen einsparen. (Das bedeutet aber nicht, dass auch Zeit gespart wurde).

Die strikte Einteilung einer Applikation in Layer und die genaue Definition, welcher Layer was zu tun hat, war anfangs sehr schwierig. Dank intensiven aber interessanten Diskussionen im Team rund um die Architektur der Applikation war diese Phase sehr lehrreich.

Nicht alle Ressourcen dürfen in JAR's gepackt werden. Der Zugriff auf Dateien ausserhalb der JAR-Datei (Konfigurationsdateien, Produktabbildungen, usw.) gestaltete sich schwieriger als angenommen. Grundsätzlich konnten wir feststellen, dass der Zugriff auf Ressourcen wie folgt vollbracht werden kann:

Für JAR-interne Ressourcen: `URL url = getClass().getResource("images/img01.jpg");`

Für JAR-externe Ressourcen: `File file = new File("images/img01.jpg");`

Diese Problematik wurde von uns zu spät erkannt, konnte aber noch im gegebenen Zeitrahmen umgesetzt werden.

Unsere Applikation ist geprägt von vielen Features, welche zwar viel Spass machen, im Endeffekt von der Aufgabenstellung aber nicht gefordert waren. Das hatte zur Folge, dass die Gruppe teilweise viel Zeit in unwichtige Dinge investierte, währenddessen die gewünschten Anforderungen beinahe vergessen gingen. Anhand unserer Aufwandschätzung (Seite 10) ist zu sehen, dass wir über 25% Mehraufwand geleistet haben. Um diesen Aufwand (und die damit verbundenen Kosten) zu optimieren, ist es wichtig, zu Beginn des Projekts genügend Zeit in die Analyse der Aufgabenstellung zu investieren.

8.2 Freigabeprotokoll

Die hier dokumentierte Durchführung des Projekts und das Ergebnis erfüllen die Erwartungen des Auftraggebers vollumfänglich. Das dokumentierte Ergebnis gilt mit Unterzeichnung als abgenommen.

Projektübergabe am **3. Mai 2011** in **Horw**.

Jürg Hofstetter
Auftraggeber

Gisler Roland
Projektbetreuer

Thomas Olnhoff
Projektbetreuer

Galliker Thomas
Studiengang Informatik (BB)

Gasser Martin
Studiengang Informatik (BB)

Stocker Elias
Studiengang Informatik (BB)
