

Abschlussbericht

Team 3:

Daniel Jäger (01518638)

Dominik Kuen (01518533)

Thomas Perteneder (00918202)

Proseminargruppe:

2

Datum:

22. Juni 2017

Inhaltsverzeichnis

1 Projektablauf	3
1.1 Geschätzter Projektaufwand und tatsächlich geleistete Stunden	3
1.2 Geplante Meilensteine und tatsächlicher Verlauf des Projekts	3
1.3 Geleisteter Arbeitsaufwand in Bezug auf Entwurfsaktivitäten	4
2 Implementierte System	5
2.1 Stabilität der Klassenstruktur	5
2.2 Stabilität der SW-Architektur	5
2.3 Funktionsumfang des Systems	5
3 Ursachenforschung	6
4 Entwicklungsumgebung	7
5 Feedback zur Proseminar-Organisation	8
5.1 Fairness	8
5.2 Bezug zur Vorlesung	8
5.3 Projektvorgaben	8
5.4 Bezug zur beruflichen Praxis	9

1 Projektablauf

1.1 Geschätzter Projektaufwand und tatsächlich geleistete Stunden

Für die Durchführung des Projekts wurde von der LV-Leitung ein wöchentlicher Arbeitsaufwand von 11.5 Stunden pro Teammitglied veranschlagt (175 Stunden insgesamt). Da kein Protokoll über die genauen Arbeitszeiten geführt wurde, lässt sich der tatsächlich geleistete Arbeitsaufwand pro Teammitglied nur schwer rekonstruieren. Zudem variierte natürlich der geleistete Arbeitsaufwand wöchentlich und pro Teammitglied.

Grundsätzlich ist zu sagen, dass es Wochen gab, wo der geforderte Arbeitsaufwand eingehalten bzw. sogar überschritten wurde und Wochen, wo dieser unterschritten wurde.

Im Schnitt wurden wahrscheinlich zwischen 5 und 15 Stunden pro Woche und Teammitglied investiert.

1.2 Geplante Meilensteine und tatsächlicher Verlauf des Projekts

Der Projektverlauf wurde in drei große Meilensteine gegliedert, die sich wiederum aus kleineren Meilensteinen zusammensetzten. Es konnten alle Meilensteine zeitgerecht abgeschlossen werden, sodass es zu keinen Verzögerungen im Projektverlauf kam.

1.3 Geleisteter Arbeitsaufwand in Bezug auf Entwurfsaktivitäten

Der wesentliche Anteil des Gesamtaufwandes wurde für die Implementierung und Entwicklung des Systems, in etwa 90%, aufgewendet. Dieser verteilt sich wiederum auf folgende Bereiche:

- Model (inkl. Repositories und Services): 35%
- Views/Controller (inkl. GUI-Gestaltung): 40%
- Tests: 15%

Auf die Planung und den Entwurf des Systems entfallen hingegen nur etwa 10% des Gesamtaufwandes. Die Planung umfasste die nachfolgend aufgelisteten Komponenten (genaue prozentuale Zuteilung nicht möglich):

- Konzeption der Klassen-/Komponentenstruktur
- Erarbeitung von Use-Cases
- Gestaltung des GUI-Prototypen
- Erstellung des Projektplans

Der eher geringe Umfang der Planungsphase ist nicht verwunderlich, wenn man in Betracht zieht, dass sich aufgrund der geringen Vorerfahrungen mit den verwendeten Frameworks viele Probleme und deren Lösungen erst in der praktischen Auseinandersetzung (Implementierungsphase) ergaben.

2 Implementierte System

2.1 Stabilität der Klassenstruktur

Die im Konzept erarbeitete Klassenstruktur wurde anfänglich komplett auf das System übertragen und bildet folglich auch den Kern des tatsächlich implementierten Datenmodells. Im Laufe der Entwicklung wurden diese ursprünglichen Klassen nur minimal adaptiert (z.B.: Hinzufügen neuer Attribute). Allerdings wurden im Entwicklungsverlauf einige weitere Klassen dem Datenmodell hinzugefügt, um die geforderten Funktionen implementieren zu können. Der Grund für das Hinzufügen weiterer Klassen lag zumeist in der leichteren Handhabung von Beziehungen und deren Abbildung auf die darunter liegenden Tabellen. Beispielsweise dient die Klasse *"ContactFor"* der Handhabung der n:m Beziehung zwischen Bezugspersonen und Kindern. Diese Vorgehensweise wurde an einigen Stellen verwendet. Da der Kern der Klassenstruktur über die gesamte Zeit stabil war, konnten die Erweiterungen problemlos durchgeführt werden.

2.2 Stabilität der SW-Architektur

Die SW-Architektur, welche auf einem Model-View-Controller Pattern fußt und Daten in einer MySQL-Datenbank persistiert, war während der gesamten Entwicklung völlig stabil. Bei der Implementierung wurde stets darauf geachtet, dass alle Funktionen sauber in der jeweiligen Schicht gekapselt sind und alle Abläufe (von der View bis zum Datenbankaufruf) transparent nachvollziehbar sind.

2.3 Funktionsumfang des Systems

Der geplante Funktionsumfang konnte mit Ausnahme der ursprünglich vorgesehenen Sonderfeatures, welche eine Bildergalerie und ein Messenger-System umfassten, vollständig umgesetzt werden. Grund hierfür ist der Ausfall zweier Teammitglieder, die das Proseminar verlassen mussten.

3 Ursachenforschung

Der gesamte Entwicklungsverlauf hat im Wesentlichen sehr gut funktioniert, sodass wir mit dem erreichten Endprodukt zufrieden sind. Deshalb können wir an dem eigentlichen Verlauf kaum etwas aussetzen. Grundsätzlich haben wir uns für ein sehr offenes Modell entschieden. Jedes Teammitglied hatte zwar einen fixen Bereich (Controller, Repositories, GUI, ...) konnte sich innerhalb dieses aber relativ frei entfalten.

Dennoch gibt es den ein oder anderen Aspekt, den wir anders handhaben würden.

Nachfolgend eine Liste:

- Probleme stets "optimal" lösen (keine unsauberen temporären Zwischenlösungen)
- Namensgebung in GUI von Beginn an sauber bzw. am geplanten Endresultat orientiert (enormer Änderungsbedarf am Ende, kann auch Tests betreffen)
- Lokale Implementierung in mehreren Klassen (z.B. Controller) absolut vermeiden
- Umgang mit Datumswerten: Mixtur aus Date, Calendar und Time vermeiden
- Frameworks (Spring, Primefaces, etc.) in allen Bereichen optimal ausnutzen (völliger Verzicht auf Eigenlösungen)
- Controller so kompakt wie möglich halten (Funktionalität (Berechnungen, etc.) immer auslagern)
- Keine Berechnungen in Services, welche im Repository realisiert werden können

4 Entwicklungsumgebung

Für die Implementierung wurden im Wesentlichen gängige Entwicklungsumgebungen („Eclipse“) und Editoren eingesetzt. Bei der Erstellung von Diagrammen kamen auch Diagramm-Editoren („yED Graph Editor“) bzw. Plugins („ObjectAid UML“) für Eclipse zum Einsatz. Alle Teammitglieder hatten im Umgang mit den genutzten Tools keinerlei Probleme und konnten die für sie optimale Entwicklungsumgebung wählen.

5 Feedback zur Proseminar-Organisation

Grundsätzlich fanden wir die Art der Projektdurchführung sinnvoll, da es sicherlich eine gute Vorbereitung für die spätere Praxis ist. Allerdings gibt es einige Aspekte, die bei einer solchen Proseminar-Organisation zu Problemen führen können. Nachfolgend unsere Meinungen zu einigen dieser.

5.1 Fairness

Die Problematik bei Team-Projekten im Allgemeinen ist, dass die Leistungen des Einzelnen nicht genau nachvollzogen werden können. Zudem ist die Gesamtleistung natürlich von allen Teammitgliedern abhängig. Kurz gesagt, es hängt sehr viel davon ab, welches Team man wählt bzw. in unserem Fall, in welches Team man zugeteilt wird. Wir mussten unser Projekt im Wesentlichen mit 3 von ursprünglich 5 Personen durchführen. Andere Teams verfügen sogar über 6 Personen. Fairness in Bezug auf Arbeitsaufwand scheint nicht gewährleistet zu sein.

5.2 Bezug zur Vorlesung

Natürlich hat das Proseminar ganz allgemein betrachtet einen starken Bezug zur zugehörigen Vorlesung. Beide beschäftigen sich mit der Entwicklung von Software. Das PS dient als Praxis, die VO liefert die Theorie. Allerdings konzentriert sich in der Praxisarbeit (im PS), vor allem, wenn man noch fast keine Projekte (außer Mini-Projekt in EVS) mit Spring, JSF und Primefaces umgesetzt hat, alles auf die Implementierungsphase. Viele Aspekte der VO (Planung, Qualität, ...) stehen im Hintergrund und spielen eine untergeordnete Rolle. Das Proseminar wirkt für den Teilnehmer deshalb völlig von der Vorlesung entkoppelt. Versteht man das PS als begleitende LV zur Vorlesung, dann wurde das Ziel eher verfehlt.

5.3 Projektvorgaben

Die PS-Leitung gibt die Thematik, Funktionsumfang und den Technologie-Stack vor, überlässt die konkrete Umsetzung und die Organisation den Teams selbst. Dies sind sinnvolle Einschränkungen, da so der Arbeitsaufwand und auch die Qualität der Projekte

(wenn Teamgröße identisch) gut vergleichbar sind, obwohl Spring nicht bei allen auf große Beliebtheit stößt.

5.4 Bezug zur beruflichen Praxis

Die Arbeit an Softwareprojekten in Teams stellt das tägliche Brot eines Entwicklers dar. Da die primäre Zielsetzung der LV aus unserer Sicht die Vermittlung eines vertieften Einblicks in diese berufliche Praxis ist, erscheint der gewählte Modus sicherlich passend. Dennoch sind die oben genannten Fallstricke zu berücksichtigen.

Die gewählte Proseminar-Organisation hat wie beschrieben Vor- und Nachteile. Alles in allem ist die Art der Proseminar-Organisation aber eine interessante Alternative zu anderen LV-Modi.