

Project Flip 20

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2011

Autoren: Lukas Elmer, Christina Heidt, Delia Treichler

Betreuer: Prof. Dr. Markus Stolze

Projektpartner: Zühlke Engineering AG, Schlieren

I. Verzeichnisse

I.1 Inhaltsverzeichnis

I. Verzeichnisse	1
I.1 Inhaltsverzeichnis.....	1
I.2 Abbildungsverzeichnis	4
I.3 Tabellenverzeichnis.....	6
II. Abstract	7
II.1 Abstract	8
III. Management Summary.....	9
III.1 Management Summary	10
IV. Extended Management Summary.....	11
IV.1 Dokumentinformationen.....	12
IV.2 Ausgangslage	13
IV.3 Vorgehen & Technologien	14
IV.4 Ergebnisse.....	15
IV.5 Ausblick.....	16
V. Technischer Bericht.....	17
V.1 Einleitung	18
V.2 Projekt Management.....	19
V.2.1 Dokumentinformationen	20
V.2.2 Projektplan	21
V.2.3 Projektorganisation	23
V.2.4 Risiken	24
V.2.5 Vorgehensmodell	25
V.3 Vorstudie	26
V.3.1 Dokumentinformationen	27
V.3.2 Vision	28
V.3.3 Vorgänger: Project Flip 1.0	29
V.3.4 Interviews.....	30
V.3.5 Personas	32
V.4 Anforderungen	37
V.4.1 Dokumentinformationen	38
V.4.2 Tools	39
V.4.3 Funktionale Anforderungen	40
V.4.4 Nichtfunktionale Anforderungen	44
V.4.5 Design Constraints.....	46

V.4.6	Zugänglichkeit (Accessibility)	47
V.5	Domain Analyse	50
V.5.1	Dokumentinformationen	51
V.5.2	Daten	52
V.5.3	Graphical User Interface (GUI)	55
V.6	Entwurf	75
V.6.1	Dokumentinformationen	76
V.6.2	Design Entscheide	77
V.6.3	Architektur	79
V.6.4	Patterns	97
V.7	Prototype Perspective Wall	98
V.7.1	Dokumentinformationen	99
V.7.2	Übersicht	100
V.7.3	Ansatz	101
V.7.4	Umsetzung	103
V.7.5	Weiterentwicklung	106
V.7.6	Fazit	107
V.8	Realisierung & Tests	108
V.8.1	Dokumentinformationen	109
V.8.2	Tests	110
V.8.3	Betriebs- & Installationsdokumentation	132
V.8.4	Code Dokumentation	134
V.8.5	Code Reviews	141
V.9	Schlussfolgerung	143
V.9.1	Dokumentinformationen	144
V.9.2	Ergebnisse	145
V.9.3	Ausblick	147
VI.	Projekt Retrospektive	148
VI.1	Dokumentinformationen	149
VI.2	Methoden und Technologien	150
VI.3	Persönliche Berichte	151
VI.3.1	Lukas Elmer	151
VI.3.2	Christina Heidt	152
VI.3.3	Delia Treichler	153
VI.4	Aufwandanalyse	154
VI.4.1	Sprints	154
VI.4.2	Personenaufwand	161
VI.4.3	Tätigkeiten	162

VI.4.4	Arbeitslisten	163
VII.	Anhang.....	173

I.2 Abbildungsverzeichnis

Abbildung 1 - Project Notes	13
Abbildung 2 - Microsoft Surface 2.....	14
Abbildung 3 - Filp 1.0	29
Abbildung 4 - Startscreen, original.....	47
Abbildung 5 - Startscreen, Deutanopie.....	48
Abbildung 6 - Startscreen, Protanopie	48
Abbildung 7 - Startscreen, Tritanopie	49
Abbildung 8 - Prozessdiagramm.....	53
Abbildung 9 - Domain Model Daten.....	54
Abbildung 10 - Objekt Diagramm Project Note	54
Abbildung 11 - Allgemeiner Ablauf	56
Abbildung 12 - Übersicht Tree View.....	56
Abbildung 13 - Tree View Stufe 1.....	56
Abbildung 14 - Tree View Stufe 2.....	57
Abbildung 15 - Tree View, Variante A1	57
Abbildung 16 - Tree View, Variante A2	58
Abbildung 17 - Tree View, Variante B	59
Abbildung 18 - Spider View	60
Abbildung 19 - Multi-Filter	61
Abbildung 20 - Finder	62
Abbildung 21 - Detailansichten	63
Abbildung 22 - Papierprototyp Variante A, Spider View	63
Abbildung 23 - Papierprototyp Variante B	65
Abbildung 24 - Papierprototyp Variante C, Detailansicht	66
Abbildung 25 - Papierprototyp Variante C, Übersicht.....	67
Abbildung 26 - Pivot Viewer.....	69
Abbildung 27 - Externes Design, Übersicht	71
Abbildung 28 - Externes Design, Detailansicht	72
Abbildung 29 - Systemübersicht	79
Abbildung 30 - Klassendiagramm mit Fokus Project Note / Metadata.....	81
Abbildung 31 - Generierung Dependency Graph	81
Abbildung 32 - Übersicht Abhängigkeiten	82
Abbildung 34 - Übersicht Abhängigkeiten mit Details	83
Abbildung 33 - Details Abhängigkeiten	83
Abbildung 35 - NDepend Dependency Graph	84
Abbildung 36 - Übersicht Interfaces.....	84
Abbildung 37 - Converter.Interfaces	85
Abbildung 38 - Services.Interfaces	85
Abbildung 39 - Loader.Interfaces	86
Abbildung 40 - Implementation Übersicht	86
Abbildung 41 - Services.Loader	86
Abbildung 42 - Converter.Pdf.....	87
Abbildung 43 - Preparer	87
Abbildung 44 - Sequenzdiagramm Preparer	88
Abbildung 45 - Services	89
Abbildung 46 - UserInterface.Surface	90
Abbildung 47 - ProjectFlip	91
Abbildung 48 - Übersicht.....	92
Abbildung 49 - Filter View	93
Abbildung 50 - Gefilterte Übersicht	93
Abbildung 51 - Info View	94
Abbildung 52 - Detail View.....	95
Abbildung 53 - Detail View mit Zoom	95
Abbildung 54 - Beispiel einer Perspective Wall, [chi91].....	101
Abbildung 55 - Skizze 3D ohne Krümmung, 90 Grad gedreht	102

Abbildung 56 - Skizze 3D mit Krümmung	102
Abbildung 57 - Skizze 3D mit X, Y	103
Abbildung 58 - Dependency Diagram 3D Komponenten	104
Abbildung 59 - Vollbild Ansicht	105
Abbildung 60 - Unit Tests 31.10.2011	110
Abbildung 61 - Test Coverage 31.10.2011	110
Abbildung 62 - Unit Tests 15.11.2011	111
Abbildung 63 - Test Coverage 15.11.2011	111
Abbildung 64 - Unit Tests 28.11.2011	112
Abbildung 65 - Test Coverage 28.11.2011	112
Abbildung 66 - Unit Tests 11.12.2011	113
Abbildung 67 - Test Coverage 11.12.2011	114
Abbildung 68 - Unit Tests 16.12.2011 Teil 1	115
Abbildung 69 - Unit Tests 16.12.2011 Teil 2	115
Abbildung 70 - Test Coverage 16.12.2011	116
Abbildung 71 - Usability nach ISO 9241-11 und Quesenberry	123
Abbildung 72 - Effektivität	128
Abbildung 73 - Effizienz	129
Abbildung 74 - Fehlertoleranz	129
Abbildung 75 - Zufriedenheit	130
Abbildung 76 - Erlernbarkeit	130
Abbildung 77 - Design	131
Abbildung 78 - Testabdeckung ohne ausgeblendete Klassen	134
Abbildung 79 - Testabdeckung mit ausgeblendeten Klassen	134
Abbildung 80 - Übersicht Metriken Visual Studio	135
Abbildung 81 - Übersicht Metriken NDepend	136
Abbildung 82 - Assemblies Metrics NDepend	136
Abbildung 83 - Keine Fehler oder Warnungen beim Kompilieren	137
Abbildung 84 - Keine Warnungen bei der Ausführung aller User Stories	138
Abbildung 85 - Naming Style	139
Abbildung 86 - Braces Layout	139
Abbildung 87 - Line Breaks and Wrapping	139
Abbildung 88 - Code Clean Up Einstellungen	140
Abbildung 89 - Laufzeitwarnungen	142
Abbildung 90 - Zeitlicher Verlauf Planung / Schätzung	155
Abbildung 91 - Aufteilung Personenaufwand	161
Abbildung 92 - Personenaufwand pro Sprint	162
Abbildung 93 - Aufwand in Stunden nach Tätigkeit	162
Abbildung 94 - Aktivitäten nach Personen gruppiert	163

I.3 Tabellenverzeichnis

Tabelle 1 - Scrum Elemente und Anwendung	25
Tabelle 2 - Tools	39
Tabelle 3 - User Stories	43
Tabelle 4 - Gegebene Daten im Zusammenhang mit den Project Notes	55
Tabelle 5 - Nutzwertanalyse.....	78
Tabelle 6 - Beispiel Mapping Dictionary.....	80
Tabelle 7 - Beispiel Mapping Datei.....	80
Tabelle 8 - System Tests Sprint 3.....	117
Tabelle 9 - System Tests Sprint 4.....	118
Tabelle 10 - System Tests Sprint 5	119
Tabelle 11 - System Tests Sprint 6	120
Tabelle 12 - System Tests Sprint 7	121
Tabelle 13 - Durchführungen Usability Tests	123
Tabelle 14 - Code Reviews mit externen Personen.....	141
Tabelle 15 - Aufwand Übersicht.....	154
Tabelle 16 - Tickets SP 0	155
Tabelle 17 - Tickets SP 1	156
Tabelle 18 - Tickets SP 2	157
Tabelle 19 - Tickets SP 3	158
Tabelle 20 - Tickets SP 4	158
Tabelle 21 - Tickets SP 5	159
Tabelle 22 - Tickets SP 6	160
Tabelle 23 - Tickets SP 7	161
Tabelle 24 - Übersicht Personenaufwand	161
Tabelle 25 - Arbeitsliste Lukas Elmer	166
Tabelle 26 - Arbeitsliste Christina Heidt.....	169
Tabelle 27 - Arbeitsliste Delia Treichler	172

Project Flip 20

Studienarbeit

II. Abstract

II.1 Abstract

Durch den Surface 2 von Microsoft ergeben sich völlig neue Möglichkeiten, Informationen interaktiv zu präsentieren. Dies möchte die Zühlke Engineering AG für die Visualisierung ihrer Projekte nutzen. Momentan stehen die Projektinformationen (nachfolgend als Project Note bezeichnet) in Papierform im Wartebereich zur Verfügung. Dabei ergibt sich einerseits das Problem, dass nie alle Projekte zur gleichen Zeit ausgestellt werden können und anderseits erschwert sich die Suche nach spezifischen Inhalten.

Diese Nachteile möchte die Zühlke Engineering AG durch die Verwendung einer Surface Applikation beheben. Zudem möchte sie damit auch ihre Expertise im Bereich Clienttechnologien untermauern und Kunden die Möglichkeiten von Microsoft Surface als Ausstellungsgerät aufzeigen.

Als Ausgangslage diente Project Flip 1.0, welches für ein Dell Multi-Touch Tablet implementiert wurde. Dieses Projekt war jedoch primär darauf ausgelegt, bei Kundengesprächen einfacher auf Projekte zugreifen zu können. Die in diesem Projekt erworbenen Erkenntnisse konnten aber für Project Flip 2.0 wichtige Informationen liefern. Um die daraus erkannten Probleme umgehen zu können, wurde für Project Flip 2.0 gleich zu Beginn ein Prototyp ausgearbeitet. Für die Entwicklung dieses Projektes wurde ein benutzerorientiertes Vorgehen gewählt. Die aus den Interviews abgeleiteten Anforderungen wurden dann in der Entwicklung umgesetzt. Der Fokus dieser Arbeit war die Erarbeitung einer einfachen, intuitiven Anwendung, welche gleich zu Beginn ohne Hilfe bedient werden kann.

Obwohl für alle Teammitglieder WPF und .NET Neuland darstellten, ist es gelungen, einen funktionstüchtigen und ansprechenden Prototyp zu entwickeln. Da der Surface 2 zum Zeitpunkt des Projektes noch nicht erhältlich war, ist die Installation und Inbetriebnahme der Applikation auf dem Gerät durch die Zühlke Engineering AG noch ausstehend.

Neben dem Client-Teil wurde auch eine Anbindung an den Server umgesetzt, über welche die Project Notes heruntergeladen werden können.

Auch wurde das User Interface durch Papierprototypen und abschliessend durch einen Usability Test durch die Benutzer validiert. Zudem wurde ein externes Design erarbeitet, um ein Konzept für die Gestaltung des User Interfaces zu definieren.

Zu jeder Project Note existieren Metadaten, welche die Project Note mit verschiedenen Begriffen umschreibt. In dieser Arbeit wurde eine Technik zur Vereinfachten Darstellung von Metadaten entwickelt. Dies war notwendig, um die grosse Menge von Metadaten auf ein für Besucher handhabbares Mass zu reduzieren. Des Weiteren setzte sich das Team mit fortgeschrittenen Visualisierungskonzepten auseinander, indem sie einen WPF Prototyp für die Perspective Wall Visualisierung entwickelte. Da die Fertigstellung einer solchen Lösung jedoch den Projektaufwand überschritten hätte, konnte sie nicht in das Projekt integriert werden.

Project Flip 20

Studienarbeit

III. Management
Summary

III.1 Management Summary

Wer einen Termin vereinbart, legt meist Wert darauf, pünktlich zu erscheinen. Dies führt oft dazu, dass man zu früh erscheint und schliesslich noch eine Weile auf seinen Gesprächspartner warten muss. Wie kann diese Zeit nun optimal genutzt werden?

Die Zühlke Engineering AG hat es sich zur Aufgabe gemacht, ihren Kunden in dieser Zeit die Firma und ihre bisher ausgeführten Projekte näher zu bringen. Daher sind in der Eingangshalle der Zühlke Engineering AG verschiedene Stellwände mit einer kleineren Auswahl an Projekten ausgestellt. Die einzelnen Projekte werden mittels einer sogenannten Project Note dargestellt – eine A4-Seite, welche Aufschluss über die Projektaufgabe, dessen Umsetzung, den Projektpartner und den Kundennutzen des Projektes gibt.

Durch die Präsentation einzelner Projekte kann sich der wartende Kunde ein besseres Bild über die Tätigkeiten der Firma machen. Er erfährt beispielsweise, in welchen Themenbereichen die Zühlke Engineering AG schon gearbeitet hat oder mit welchen Partnern sie dies tat. Dadurch könnten auch Anregungen für eine zusätzliche Zusammenarbeit entstehen.

Des Weiteren gibt es die Möglichkeit, von den aufgeföhrten Project Notes eine Kopie mitzunehmen. Diese Kopien liegen zur Mitnahme bereit oder können gegebenenfalls am Empfang in Auftrag gegeben werden.

Das hauptsächliche Problem besteht darin, dass nie alle Projekte gleichzeitig ausgestellt werden können. Es ist denkbar, dass sich Personen für Projekte oder Bereiche interessieren, die zum Zeitpunkt nicht aufgeführt sind. Dies kann von den Kunden auch als Fehlen solcher Projekte interpretiert werden.

Eine weitere Problematik ist die Zeit, welche für das Durchschauen der Project Notes benötigt wird, wenn z.B. nach einem bestimmten Themenbereich oder einer bestimmten Firma gesucht wird. Diese ist nur begrenzt und könnte besser genutzt werden.

Die Zühlke Engineering AG hat ihre Stärken unter anderem im Bereich des Produkt- und Software-Engineerings. Durch Project Flip 2.0 kann sie ihre Expertise in diesem Gebiet bestens unter Beweis stellen. Der Microsoft Surface 2, welcher zur Realisierung einer progressiven Lösung für die bestehende Problematik eingesetzt wird, zeigt, dass die Firma innovativ und bezüglich Technologien auf dem neusten Stand ist. Zudem bietet das Endprodukt eine interaktive und originelle Möglichkeit, Kunden einerseits über die Zühlke Engineering AG zu informieren und andererseits zu unterhalten. Es wird möglich sein, während einer kurzen Wartezeit einfach und auf spielerische Weise Wissen zu sammeln – dies dank intuitiver und unproblematischer Bedienung des Gerätes.

Das Produkt soll den Benutzer dazu animieren, neue Seiten der Zühlke Engineering AG zu entdecken. Zudem erlaubt es, in bestimmten Themenbereichen zu stöbern. Dadurch kann beispielsweise erfahren werden, mit welchen Technologien die Firma bereits gearbeitet hat oder für welchen anderen Auftraggeber die Zühlke Engineering AG erfolgreich Projekte ausgeführt hat.

Ist man schliesslich auf die gewünschten Themen gestossen, bietet sich die Möglichkeit, die dieser Thematik untergeordneten Project Notes per Email zu versenden oder direkt ausdrucken zu lassen.

Project Flip 2.0

Studienarbeit

IV. Extended
Management
Summary

IV.1 Dokumentinformationen

Datum	Version	Änderung	Autor
13.12.2011	1.0	Erste Version des Dokuments	cheidt
18.12.2011	1.1	Review	dreichl
20.12.2011	1.2	Korrekturen Besprechung, Bilder	cheidt

IV.2 Ausgangslage

Wer einen Termin vereinbart, legt meist Wert darauf, pünktlich zu erscheinen. Dies führt oft dazu, dass man zu früh erscheint und schliesslich noch eine Weile auf seinen Gesprächspartner warten muss. Wie kann diese Zeit nun optimal genutzt werden?

Die Zühlke Engineering AG hat es sich zur Aufgabe gemacht, ihren Kunden in dieser Zeit die Firma und ihre bisher ausgeführten Projekte näher zu bringen. Daher sind in der Eingangshalle der Zühlke Engineering AG verschiedene Stellwände mit einer kleineren Auswahl an Projekten ausgestellt. Die einzelnen Projekte werden mittels einer sogenannten Project Note dargestellt – eine A4-Seite, welche Aufschluss über die Projektaufgabe, dessen Umsetzung, den Projektpartner und den Kundennutzen des Projektes gibt (siehe Abbildung 1 - Project Notes).

Durch die Präsentation einzelner Projekte kann sich der wartende Kunde ein besseres Bild über die Tätigkeiten der Firma machen. Er erfährt beispielsweise, in welchen Themenbereichen die Zühlke Engineering AG schon gearbeitet hat oder mit welchen Partnern sie dies tat. Dadurch könnten auch Anregungen für eine zusätzliche Zusammenarbeit entstehen.

Des Weiteren gibt es die Möglichkeit, von den aufgeführten Project Notes eine Kopie mitzunehmen. Diese Kopien liegen zur Mitnahme bereit oder können gegebenenfalls am Empfang in Auftrag gegeben werden.

Das hauptsächliche Problem besteht darin, dass nie alle Projekte gleichzeitig ausgestellt werden können. Es ist denkbar, dass sich Personen für Projekte oder Bereiche interessieren, die zum Zeitpunkt nicht aufgeführt sind. Dies kann von den Kunden auch als Fehlen solcher Projekte interpretiert werden. Eine weitere Problematik ist die Zeit, welche für das Durchschauen der Project Notes benötigt wird, wenn z.B. nach einem bestimmten Themenbereich oder einer bestimmten Firma gesucht wird. Diese ist nur begrenzt und könnte besser genutzt werden.



Abbildung 1 - Project Notes

IV.3 Vorgehen & Technologien

Für diese Arbeit war der Microsoft Surface 2¹ (siehe Abbildung 2 - Microsoft Surface 2) als Hardware schon fest vorgegeben.

Dadurch war auch die Technologie schon von Anfang an auf .NET und WPF beschränkt.

Als Ausgangslage diente Project Flip 1.0, welches für ein Dell Multi-Touch Tablet implementiert wurde. Dieses Projekt war jedoch primär darauf ausgelegt, bei Kundengesprächen einfacher auf Projekte zugreifen zu können. Die in diesem Projekt erworbenen Erkenntnisse konnten aber für Project Flip 2.0 wichtige Informationen liefern. Um die daraus erkannten Probleme umgehen zu können, wurde für Project Flip 2.0 gleich zu Beginn ein Prototyp ausgearbeitet.

Im Fokus dieser Applikation liegt klar der Nutzer. Deshalb wurden anfangs Befragungen zum Verhalten und den Gewohnheiten der Kunden der Zühlke Engineering AG durchgeführt. Die daraus abgeleiteten Nutzerprofile bildeten den Grundstein der Anwendung. Zudem musste bedacht werden, dass die Nutzung der Applikation durch Kunden immer nur von kurzer Dauer ist. Aus diesem Grund offeriert die Applikation nur eine begrenzte Anzahl von Funktionen. Diese umfassen das Stöbern nach Project Notes, dass Filtern nach verschiedenen Kriterien und schliesslich das Lesen einer Project Note.

Nach der Auswertung der Interviews und Beobachtungen wurden verschiedene Möglichkeiten der Informationsvisualisierung in Papierform aufgezeichnet. Diese Prototypen wurden mit Testpersonen, die den Nutzerprofilen entsprechen, getestet und die Ergebnisse ausgewertet. Anschliessend wurde mit der Implementierung der Funktionalitäten begonnen.

Zu jeder Project Note existieren Metadaten, welche die Project Note mit verschiedenen Begriffen umschreibt. Diese wurden für den Filter als Einschränkungskriterien aus den Project Notes extrahiert. Da diese sehr umfangreich sind, würde der Nutzer aber schnell die Übersicht verlieren. Daher sollen ähnliche Begriffe zu einem zusammengefasst und so der Umfang reduziert werden. Es wurde eine Möglichkeit erarbeitet, die Metadaten dynamisch zu aggregieren und einfach zu warten. Dazu wurde eine Variante gewählt, bei welcher über ein Textdokument die verschiedenen Begriffe auf einen Begriff abgebildet werden können. Die Datei kann auch problemlos von projektaussenstehenden Personen bearbeitet werden, was die Wartbarkeit sichert.

Die Zühlke Engineering AG verfügt über ein Corporate Design, welches Farb- und Schriftpaletten umfasst. Diese wurden verwendet, um ein externes Design zu erstellen. Einerseits wurde so ein Rahmen für die Umsetzung der Benutzeroberfläche gesetzt und andererseits ist die Anwendung so auf den ersten Blick als eine Zühlke Engineering AG Applikation erkennbar.

Des Weiteren setzte sich das Team mit fortgeschrittenen Visualisierungskonzepten auseinander, indem sie in Grundzügen eine Perspective Wall programmierte. Da die Fertigstellung einer solchen Lösung jedoch den Projektaufwand überschritten hätte, konnte sie nicht in das Projekt integriert werden.

Um die Interaktion mit dem Microsoft Surface 2 und das Verständnis für den Programmverlauf zu vereinfachen, kommen, neben den funktionalen Anforderungen, auch den Animationen und der Verwendung von Gesten eine grosse Bedeutung zu. Diese sind Begeisterungsfaktoren und sollen den Nutzern die Bedienung so angenehm wie möglich gestalten. Daher wurde in diesem Projekt für die Umsetzung dieser Faktoren Zeit eingeräumt. Der zeitliche Aufwand war grösser als erwartet. Daher mussten gewisse gewünschte Funktionen vereinfacht gelöst werden.

Abschliessend wurden mit verschiedenen Testpersonen Usability Tests durchgeführt, um die Benutzeroberfläche zu prüfen. Die von den Testpersonen erhaltenen Anregungen konnten gegen Ende des Projektes noch umgesetzt werden.



Abbildung 2 - Microsoft Surface 2

¹ Bildquelle: [http://www.samsunglfd.com/upload/product/img/Surface\[1294390605622product\].jpg](http://www.samsunglfd.com/upload/product/img/Surface[1294390605622product].jpg), letzter Zugriff: 20.12.2011

IV.4 Ergebnisse

Obwohl für alle Teammitglieder WPF und .NET Neuland darstellten, ist es gelungen, einen funktionstüchtigen und ansprechenden Prototyp zu entwickeln. Das Endprodukt bietet eine interaktive und originelle Möglichkeit, Kunden einerseits über die Zühlke Engineering AG zu informieren und andererseits zu unterhalten.

Der Prototyp erfüllt die wichtigsten Anforderungen an das Projekt. Diese umfassen eine ansprechende Darstellung der Project Notes, eine intuitive Navigation und Suche sowie die Filterung der Project Notes nach verschiedenen Kriterien.

Die einfache Nutzung der Applikation wurde durch das Testen der Benutzeroberfläche durch Testpersonen sichergestellt.

Neben dem Client-Teil wurde auch eine Anbindung an den Server umgesetzt, über welche die Project Notes heruntergeladen werden können.

Die Vorteile zur existierenden Lösung bestehen darin, dass nicht nur eine beschränkte Anzahl von Projekten sondern alle vorhandenen Project Notes verfügbar sind. Zudem wird die Möglichkeit geboten, nach spezifischen Inhalten zu suchen.

Project Flip 2.0 bildet eine solide, gut ausbaubare Grundlage für die Weiterentwicklung zur vollwertigen Applikation. Da der Surface 2 zum Zeitpunkt des Projektes noch nicht erhältlich war, ist die Installation und Inbetriebnahme der Applikation auf dem Gerät durch die Zühlke Engineering AG noch ausstehend.

IV.5 Ausblick

Project Flip 2.0 befindet sich noch im Entwicklungsstadium. Bevor die Applikation effektiv zum Einsatz kommen kann, müssen noch einige Arbeiten vorgenommen werden. Der wichtigste Punkt ist dabei die Anbindung an das Intranet der Zühlke Engineering AG. Weiter stellten die Animationen in diesem Projekt eine Herausforderung dar und müssen noch weiter ausgebaut werden. Gesten werden infolge des begrenzten Projektzeitrahmens noch nicht unterstützt. Auch diese müssten bei der Weiterentwicklung des Projektes noch implementiert werden.

Der Prototyp erfüllt nur die wichtigsten Anforderungen an das Projekt. Weitere Anforderungen an die Software sind die Anbindung an das Intranet der Zühlke und das Verschicken der Project Notes via Email. Dazu wäre auch das Scannen von Visitenkarten und die Verarbeitung der so gewonnenen Daten denkbar. Auch eine Volltextsuche würde sich als sinnvoll erweisen.

Diese Anforderungen konnten nicht umgesetzt werden. Ein Grund dafür ist auch die Verschiebung des Erscheinungsdatums des Microsoft Surface 2 auf Februar 2012.

Project Flip 20

Studienarbeit

V. Technischer Bericht

V.1 Einleitung

Der technische Bericht beginnt mit dem Projektmanagement (siehe V.2 Projekt Management), in welchem der Projektplan dargelegt und die Projektorganisation umschrieben wird. Des Weiteren werden die Risiken und das gewählte Vorgehensmodell aufgezeigt.

Das nächste Kapitel enthält die Vorstudie (siehe V.3 Vorstudie). Diese zeigt die Vision auf, welche die existierende und die gewünschte Lösung umschreibt. Zudem wird der Vorgänger, Project Flip 1.0, beschrieben und die aus diesem Projekt gezogenen Schlüsse aufgelistet.

Danach ist das Interview, welches die Grundlage zur Erstellung der Personas war, dokumentiert. Die Personas und die dazu gehörigen Szenarien werden daraufhin vorgestellt.

Das Kapitel Anforderungen (siehe V.4 Anforderungen) geht kurz auf die in dem Projekt verwendeten Tools ein. Danach umschreibt es die funktionalen und nichtfunktionalen Anforderungen, Design Constraints und Accessibility im Zusammenhang mit der Nutzung des Surface.

Im Kapitel Domain Analyse (siehe V.5 Domain Analyse) werden zu Beginn die Daten und das zugehörige Domain Model beschrieben. Auch wird der Prozess zur Entstehung einer Project Note erklärt. Danach wird auf das Graphical User Interface (GUI) eingegangen. Dafür wurde ein Creative Workshop durchgeführt, die daraus entstandenen Ideen führten zu den dokumentierten Papier-Prototypen. Im Anschluss daran werden die Design Entscheide und das externe Design vorgestellt. Schliesslich wird aufgelistet, welche Surface Guidelines sich nicht mit den Projektanforderungen decken und daher verletzt wurden.

Das Kapitel Entwurf (siehe V.6 Entwurf) widmet sich einleitend den Design Entscheiden. Daraufhin wird die gewählte Architektur umschrieben. Dazu gehören auch die Assemblies und Namespaces. Anschliessend wird auf das User Interface Design eingegangen. Danach werden die Prozesse und Threads beschrieben und zum Schluss die verwendeten Patterns aufgelistet.

Im Kapitel Perspective Wall Prototype (siehe V.7 Prototype Perspective Wall) ist dokumentiert, wie der Prototyp für eine 3D-Ansicht der Project Notes Übersicht erstellt wurde.

Im nachfolgenden Kapitel Realisierung und Test (siehe V.8 Realisierung & Tests) sind die Unit und System Tests dokumentiert. Dann wird auf die durchgeführten Usability Tests und deren Auswertung eingegangen. Anschliessend wird die Betriebs- und Installationsdokumentation umschrieben.

Das letzte Kapitel Schlussfolgerung (siehe V.9 Schlussfolgerung) widmet sich den mit der Arbeit erzielten Ergebnissen und dem Ausblick für die Weiterentwicklung des Projektes.

Project Flip 20

Studienarbeit

V.2 Projekt
Management

V.2.1 Dokumentinformationen

Datum	Version	Änderung	Autor
23.09.2011	1.0	Erste Version des Dokuments	dreichl
30.09.2011	1.1	Kapitel Vorgehensmodell	dreichl
30.09.2011	1.2	Review	cheidt
02.10.2011	1.3	Ergänzungen Kapitel Vorgehensmodell	dreichl
14.10.2011	1.4	Anpassung Projektplan Sprint 3	dreichl
31.10.2011	1.5	Anpassung Projektplan Sprint 4	dreichl
14.11.2011	1.6	Anpassung Projektplan Sprint 5	dreichl
28.11.2011	1.7	Anpassung Projektplan Sprint 6	dreichl
16.12.2011	1.8	Anpassung Projektplan Sprint 7	cheidt

V.2.2 Projektplan

Das Management des Projektes Project Flip 2.0 geschieht im Redmine (<https://redmine.elmermx.ch/>).

V.2.2.1 Prototyp

Mit dem Prototyp wurde sichergestellt, dass es möglich ist, die Project Notes in vektorisierter Form darzustellen. Zudem wurde die grundlegende Architektur damit festgelegt.

V.2.2.2 Releases

Da das Projekt nach dem Vorgehensmodell Scrum (siehe Unterkapitel V.2.5 Vorgehensmodell) durchgeführt wird, gibt es nur zu Projektende einen Release.

Während des Projekts wird immer am Ende eines Sprints ein voll lauffähiger Prototyp vorliegen.

Folgender Release ist vorgesehen:

Version	Typ	Beschreibung	Datum
1.0	Finalversion	Fertige Version, Bugs gefixt	23.12.2011

V.2.2.3 Milestones

Ein Milestone ereignet sich jeweils am Ende eines Sprints. Die Sprints wie auch die Milestones sind im Redmine dokumentiert.

Nachfolgend eine Übersicht über die Milestones, für genauere Angaben wird auf das Redmine verwiesen. Die detaillierte Planung der jeweiligen Sprints erfolgt schrittweise nach dem Vorgehensmodell Scrum (siehe Unterkapitel V.2.5 Vorgehensmodell).

Sprint 1	Inhalt	Termine
23.09.2011 - 30.09.2011	- Entwicklungsumgebung einrichten, Einarbeitung - Interviews, Kundenanalyse - Vision, Persona, Szenarien V0 - Architektur-Prototyp	19.09.11 Kickoff-Meeting 26.09.11 Wochenreview 03.10.11 Review Sprint 1

Sprint 2	Inhalt	Termine
03.10.2011 - 17.10.2011	- Vision, Persona, Szenarien V1 - User Stories - Creative Workshop - Papierprototyp	10.10.11 Wochenreview 17.10.11 Review Sprint 2

Sprint 3	Inhalt	Termine
17.10.2011 - 31.10.2011	- Must User Stories mit Priorität 1 implementiert und getestet - Nichtfunktionale Anforderungen	24.10.11 Wochenreview 31.10.11 Review Sprint 3

Sprint 4	Inhalt	Termine
31.10.2011 - 14.11.2011	- Für diesen Sprint geplante User Stories implementiert und getestet - Layout definiert	04.11.11 Codereview mit Christian Moser 07.11.11 Wochenreview 11.11.11 Demonstration Must User Stories, Schlieren 14.11.11 Review Sprint 4

Sprint 5	Inhalt	Termine
14.11.2011 - 28.11.2011	- Für diesen Sprint geplante User Stories implementiert und getestet	25.11.11 Vorgezogene Demo der User Stories für Zühlke 28.11.11 Review Sprint 5

Sprint 6	Inhalt	Termine
28.11.2011 - 12.12.2011	- Für diesen Sprint geplante User Stories implementiert und getestet - Dokumentationen geschrieben - Codereview	05.12.11 Wochenreview 09.12.11 Demonstration der Software, Schlieren 12.12.11 Wochenreview

Sprint 7	Inhalt	Termine
12.12.2011 - 23.12.2011	- Management Summary, Abstract geschrieben - Video gedreht - Projekt abgeschlossen	19.12.11 Wochenreview 23.12.11 Projektabgabe

V.2.2.4 Zeitplan und Zeiterfassung

Die einzelnen Arbeitspakete (Tickets) sind den jeweiligen Sprints zugeordnet. Das Projekt ist in sieben Sprints unterteilt. Das Ende eines Sprints entspricht jeweils einem Milestone.

Die komplette Zeitplanung und die Zeiterfassung werden auf dem Redmine-Server durchgeführt. Für jedes Arbeitspaket wird der Zeitaufwand geschätzt und ein Ticket erstellt. Diese Tickets werden den jeweiligen Sprints zugeordnet. Wurde an einem Ticket gearbeitet, wird die dafür aufgewendete Zeit auf das Ticket gebucht. Die Erfassung der Zeit für die jeweilig bearbeiteten Tickets geschieht sofort nach Abschluss der Arbeiten. Somit ist die Zeiterfassung stets aktuell.

Die Reportfunktion bietet einen Überblick über den geplanten und tatsächlichen Zeitaufwand vermitteln. Zudem ist es möglich, den Arbeitsaufwand mittels einer cvs-Datei zu exportieren und z.B. in Excel anschaulich darzustellen.

V.2.2.5 Aufwandschätzung

Die Aufwandschätzung ergibt sich durch den geschätzten Aufwand pro Ticket im Redmine.

V.2.3 Projektorganisation

V.2.3.1 Team

V.2.3.1.1 Lukas Elmer (Abk. lelmer)

Kenntnisse in: Ruby on Rails, PHP, Python / Django, Typo3, Wordpress, Java, XHTML, JavaScript, C++, Ubuntu Server

Rolle/Verantwortlichkeiten: Architektur, Libraries, Serverunterhalt Redmine / SVN

Mailadresse : lukas.elmer@gmail.com

Skype Adresse: lukas.elmer



V.2.3.1.2 Christina Heidt (Abk. cheidt)

Kenntnisse in: Java, HTML/CSS, C++, Photoshop

Rolle/Verantwortlichkeiten: Grafisches Design, Sitzungsprotokollierung, Risikomanagement

Mailadresse: heidt.christina@gmail.com

Skype Adresse: christina_heidt



V.2.3.1.3 Delia Treichler (Abk. dtreichl)

Kenntnisse in: Java, HTML/CSS, C++

Rolle/Verantwortlichkeiten: Überwachung und Erstellung Projektplan, Teamsitzungen

Mailadresse: deliatreichler@gmail.com

Skype Adresse: de-lia



V.2.4 Risiken

Das Risikomanagement befindet sich im VII Anhang, C.

V.2.5 Vorgehensmodell

Für das Projekt Project Flip 2.0 wird hauptsächlich der Ansatz von Scrum verfolgt, weil dieses Vorgehensmodell auf die Eigenorganisation der einzelnen Teammitglieder ausgerichtet und äusserst produktiv ist, da Overhead so weit wie möglich reduziert wird.

Die nachfolgende Tabelle zeigt auf, welche Elemente (Rollen, Meetings und Artefakte) von Scrum im Projekt Project Flip 2.0 wie gehandhabt werden.

Scrum-Element	Umsetzung
Rollen	
Product Owner, Scrum Master	Diese Rollen können personalbedingt nicht besetzt werden. Die Aufgaben des Product Owners und des Scrum Masters werden vom Entwicklungsteam übernommen.
Meetings	
Sprint Planung	Die Planung des darauf folgenden Sprints wird zu Ende des gegenwärtigen Sprints durchgeführt.
Daily Scrum	Das Meeting wird jeweils montags und freitags durchgeführt (es wird kein Protokoll geführt).
Sprint Review	Das Review Meeting findet jeweils am letzten Tag des aktuellen Sprints statt.
Artefakte	
Product Backlog	Die Anforderungen an das Produkt sind als Tickets im Redmine erfasst. Die Schätzung des Aufwands geschieht für jede Anforderung auf dem entsprechenden Ticket nach dem Modell Planning Poker.
Sprint Backlog	Die für einen Sprint geplanten Aufgaben existieren im Redmine als Ticket, welche dem jeweiligen Sprint zugeordnet sind. Der Restaufwand, der für eine einzelne Aufgabe noch benötigt wird, ist über die Differenz der geschätzten und bisher gebuchten Zeit ersichtlich. Gleichzeitig dient das Redmine als Ersatz für das Taskboard.
Burndown-Charts	Das Gantt-Diagramm, welches sich im Redmine anzeigen lässt, erübrigt einen Burndown-Chart.
Impediment Backlog	Im Redmine Wiki besteht eine Seite zur Eintragung von Hindernissen und Problemen des Projektes.
Releaseplan	Die Tickets im Redmine bieten einen Überblick über den Zeitplan und die Termine/Meetings. Die erwartete Anzahl Sprints ist unter Roadmap ersichtlich. Für die Behandlung von Risiken siehe Unterkapitel V.2.4 Risiken.

Tabelle 1 - Scrum Elemente und Anwendung

Das Verfassen formaler Dokumente sowie die Erstellung eines Architekturprototyps sind dem Vorgehensmodell RUP entnommen.

Project Flip 2.0

Studienarbeit

V.3 Vorstudie

V.3.1 Dokumentinformationen

Datum	Version	Änderung	Autor
23.09.2011	1.0	Erste Version des Dokuments	cheidt
26.09.2011	1.1	Anpassungen mit Informationen aus Interview	cheidt
30.09.2011	1.2	Vorgänger: Project Flip 1.0 hinzugefügt	lelmer
03.10.2011	1.3	Review, Ergänzungen Vision	dtreichl
10.10.2011	1.4	Ergänzung 3. Persona inkl Szenarien, Interview	cheidt
10.10.2011	1.5	Review 3. Persona inkl Szenarien	dtreichl
16.10.2011	1.6	Review, kleinere Ergänzungen, Soll Szenario Persona 2	lelmer
24.10.2011	1.7	Anpassung/Ergänzung 3. Persona inkl Szenarien	dtreichl

V.3.2 Vision

Siehe Kapitel III Management Summary

V.3.3 Vorgänger: Project Flip 1.0

Gegen Ende des Jahres 2008 wurde von Mitarbeitern der Zühlke Engineering AG in einem Weiterbildungscamp die Version 1.0 des Project Flip entwickelt. Dabei handelt es sich um einen Prototyp, der Project Notes auf einem Tablet-PC anzeigt.

Die neu zu entwickelnden Version 2.0 unterscheidet sich in einigen Punkten wesentlich von der bisherigen Version.

Dieses Kapitel zeigt die Ziele, die Funktionalitäten und die Schwierigkeiten der ersten Version von Project Flip.



Abbildung 3 - Filp 1.0

V.3.3.1 Ziele

Die Version 1.0 verfolgte zwei Hauptziele. Das erste Bestreben war das Studium der dazumal neuen Touch-Technologie. Die zweite Absicht war die ansprechende Darstellung von Project Notes. Sie bietet den Verkäufern von Zühlke Engineering AG eine Möglichkeit, Project Notes an ein Verkaufsgespräch mitzunehmen und dem Kunden auf eine innovative Art zu präsentieren.

Durch die Festlegung der oben genannten Ziele war die Applikation ganz klar auf Tablet-PCs ausgerichtet.

V.3.3.2 Funktionalitäten

- Blättern, Cover Flow
- Select, Zoom
- Verschiedene Filtermöglichkeiten mit Texteingabe
- Email (per Outlook → Tablet PC)
- In externem Programm öffnen (Adobe Reader)

V.3.3.3 Lessons Learned in Version 1.0

V.3.3.3.1 Bildumwandlung

Jede Project Note, deren Originalversion ein PDF-Dokument ist, wurde in ein Bild umgewandelt und benötigte daher mehr Speicherplatz. Zusätzlich bot das dadurch entstandene Bild der Project Note eine schlechtere Lesequalität, da die darin enthaltenen Bilder und Schriften nicht vektorisiert waren.

V.3.3.3.2 Performance / Hardware

Da die Hardware des Tablet PCs auf akkuschonende Komponenten optimiert ist, stellte die Leistung des PCs ein grösseres Problem dar. Zum Beispiel flackerten die Objekte, wenn sie per Multitouch grösser oder kleiner gemacht wurden. Zusätzlich war die Auflösung des Monitors nicht optimal, was zu schlechter Lesbarkeit der Dokumente führte.

V.3.4 Interviews

Das Interview wurde mit einer der Empfangsdamen der Zühlke Engineering AG durchgeführt. Dies wurde getan, da es eher schwierig gewesen wäre, die wartenden Kunden zu befragen. Einerseits haben die meisten wenig Zeit und andererseits wäre es mit einem grossen Zeitaufwand verbunden gewesen. Das Empfangspersonal hat jedoch Zeit, die wartenden Kunden zu beobachten und wird bei Problemen auch angesprochen.

V.3.4.1 Fragen und Antworten

F1: Wie alt sind etwa die typischen Kunden?

Die meisten etwa ab 35-40. Breites Spektrum.

Meist Projektleiter, Standortchefs, Abteilungsleiter, CEO. Da gemeinsames Projekt.

Kunden kommen durch Homepage oder Telefon, weil sie bei gewissen Themen Hilfe/Kontakt brauchen.

Erster Kontakt: Business Meeting oder Veranstaltungen und Events.

Durch Events entstehen weiterer Kontakt -> führt zu neuen Projekten!

Bei erstem Besuch Projekte zeigen und herumführen.

F2: Anteil Mann/Frau?

3/4 Männer 1/4 Frauen.

F3: Wie gross ist das technische Interesse der Kunden?

Meist Leute die grosses technisches Wissen mitbringen.

Einerseits Entscheider/Geldgeber: schaut wegen Geld und Qualität, andererseits Umsetzer/Projektleiter

Gremium kommt zusammen, jedoch Anreise meist separat. Dann teilweise längere Wartezeit. Nur gesamte Gruppe wird nach oben gelassen, ansonsten warten im Eingangsbereich bis alle eingetroffen

F4: Wie lange müssen die Kunden typischerweise warten?

Meistens höchstens 5 bis 10 Minuten. Aber: Gruppenmitglieder treffen z.T. zu früh ein -> haben Zeit.

Bei längeren Sitzungen mit Businesslunch inklusive Firma zeigen.

F5: Wirken sie gestresst?

Eher selten.

F6: Sind einige davon am Telefon?

Eher selten, die meisten haben bei Meetings ihr Telefon abgeschaltet.

Bei Meeting vermutlich schon geistig in Vorbereitung zu Meeting.

F7: Wieviele davon setzen sich mit den Project Notes auseinander?

Die meisten setzen sich mit Project Note aus ihrem Bereich auseinander (Bsp. .NET). Nehmen auch oft etwas mit, fragen nach. Fragen ob man gewisse Bücher kaufen kann.

Manchmal Nachfrage ob sie gewisses Dokument zugeschickt bekommen. Lesen es oft detailliert und gezielt.

Mitnehmen des Dokuments ist auch sehr wichtig.

F8: Wieviel Zeit verbringen sie dort?

Nehmen das Gebiet, das sie interessiert.

F9: Wie verhalten sie sich vor den Project Notes? (Übersicht verschaffen, nur 2-3 anschauen)

Sehr interessiert in ihrem Bereich -> Project Note suchen und dann GENAU lesen

F10: Wieviele sitzen ab, wieviele stehen?

Die meisten lesen im Sitzen

Weiteres:

Viele schauen auch die Timeline an, Bücher über Firmengeschichte.

V.3.5 Personas

Aus dem Interview lassen sich primär zwei Personas erstellen. Die eine ist für die Umsetzung von Projekten zuständig (siehe V.3.5.1 Persona Ulrich Umsetzer) und die andere ist für den finanziellen Aspekt und die Projektabsegnung verantwortlich (siehe V.3.5.2 Persona Erik Entscheider). Gerade die erste Persona Ulrich Umsetzer ist mehr fokussiert auf Projekte, die im gleichen Themenbereich sind, in dem sie selbst auch tätig ist.

Zudem wurde von den Betreuern der Zühlke Engineering AG noch vorgeschlagen, dass eine dritte Persona erstellt wird. Hierbei handelt es sich um Personen, die sich die Zühlke Engineering AG als Arbeitgeber vorstellen könnten und daher an der Firma interessiert sind (siehe V.3.5.3 Persona Bettina Bewerberin). Gerade diese werden sich besonders bemühen, pünktlich zu erscheinen und somit Zeit haben, sich mit den Project Notes auf dem Surface 2 auseinander zu setzen.

V.3.5.1 Persona Ulrich Umsetzer

Ulrich Umsetzer	
	Kurzprofil Projektleiter 42 Jahre Ausgeprägtes technisches Verständnis und Computerkenntnisse
Arbeitskontext (Lärm, Unterbrüche, Regeln)	Bei der Arbeitsumgebung handelt es sich um die Eingangshalle der Zühlke Engineering AG. Falls Ulrich Umsetzer dort warten muss, kann er sich Zeit nehmen, um sich einen Überblick über die vorangegangenen Projekte der Firma zu verschaffen. Unterbrechungen entstehen dadurch, dass er von einem Mitarbeiter abgeholt wird oder eine Unterhaltung mit jemandem beginnt, der ebenfalls wartet.
Persönlichkeit & Vorlieben	Ulrich Umsetzer ist ein sehr interessierter und engagierter Angestellter der Beispiefirma. Insbesondere neue Technologien faszinieren ihn ungemein und Ulrich schätzt jede Möglichkeit, bei der er diese Neuheiten kennenlernen kann. Zudem ist Ulrich selbst ein begeisterter Smartphone-Besitzer. Er schätzt die vielen Gebrauchsmöglichkeiten, welche diese Geräte bieten. Als sehr angenehm empfindet er außerdem deren die intuitive Handhabung.
Vorkenntnisse & Lernen (Computer, Domain)	Ulrich arbeitet seit zehn Jahren bei der Beispiefirma. Seine Aufgabe besteht in der Leitung von Projekten im Bereich Produkt- und Software-Engineering. Er hat schon mehrmals mit der Zühlke Engineering AG zusammengearbeitet oder hat sich in gewissen Projekten von einem Experten der Firma unterstützen lassen. Aus diesem Grund ist er immer wieder neugierig zu erfahren, mit welchen neuen Themen sich die Zühlke Engineering AG auseinandergesetzt hat.
Eigenschaften / Verhaltensvariablen	<ul style="list-style-type: none"> • Technische Expertise • Kenntnisse in Produkt- und Software-Engineering • Kenntnisse über mobile Applikationen • Smartphone-Kenntnisse
Ziele	<ul style="list-style-type: none"> • Zeit überbrücken • Sich über die Zühlke Engineering AG informieren

V.3.5.1.1 Ist-Szenario-1

Ulrich Umsetzer hat sich mit zwei Mitarbeitern der Zühlke Engineering AG verabredet um eine geplante Zusammenarbeit zu besprechen. Dabei handelt es sich um ein Projekt im .Net-Bereich. Ulrich erscheint pünktlich bei der Firma, erfährt aber von der Empfangsdame, dass die beiden Mitarbeiter sich etwas verspäten. Er entscheidet sich daher, die Zeit zu nutzen, um sich bei den ausgestellten Project Notes umzusehen. Dabei sucht er nach Projekten, die ebenfalls mit .Net umgesetzt wurden. Nach kurzem Suchen stösst er auf das erste Projekt, welches diesen Kriterien entspricht. Er studiert die Project Note daher aufmerksam.

Wenig später wird er von den beiden Mitarbeitern der Zühlke Engineering AG abgeholt. Leider hat Ulrich in dieser Zeit die Project Note nicht zu Ende lesen können. Er würde daher gerne eine Kopie davon mitnehmen, doch gibt es nicht von allen ausgestellten Projekten Kopien. Ulrich bittet daher einen der beiden Mitarbeiter, ihm diese doch zukommen zu lassen. Dieser willigt gerne ein und schlägt vor, dass er Ulrich auch gleich mehrere Projekte aus diesem Bereich zukommen lassen könnte.

Nach der Besprechung begibt sich Ulrich zurück ins Büro der Beispielfirma. Dort angekommen, ruft er seine Emails ab. Der Zühlke Engineering AG Mitarbeiter hat ihm eine kleine Auswahl an spannenden .Net-Projekten zukommen lassen. Ulrich druckt zwei, die ihm besonders interessant erscheinen, aus und liest diese nun in aller Ruhe.

V.3.5.1.2 Soll-Szenario-1

Ulrich Umsetzer hat sich mit zwei Mitarbeitern der Zühlke Engineering AG verabredet um eine geplante Zusammenarbeit zu besprechen. Dabei handelt es sich um ein Projekt im .Net-Bereich. Ulrich erscheint pünktlich bei der Firma, erfährt aber von der Empfangsdame, dass die beiden Mitarbeiter sich etwas verspäten. Die Empfangsdame weist ihn zudem darauf hin, dass die Project Notes nun über den neu erworbenen Surface Tisch eingesehen werden können. Ulrich sieht dies als ideale Chance, ein solches Gerät einmal selber zu testen. Interessiert setzt er sich daher an den Tisch und beginnt sogleich nach Projekten im .Net-Bereich zu suchen. Diese findet Ulrich auch schnell und er ist beeindruckt über die grosse Auswahl an Projekten. Ulrich sucht sich davon einige interessante aus und beginnt diese zu lesen.

Wenig später wird er von den beiden Mitarbeitern der Zühlke Engineering AG abgeholt. Nun würde Ulrich die Artikel aber gerne doch genauer studieren. Er nutzt daher die Möglichkeit, dass die Project Notes auch an eine Emailadresse verschickt werden können:

- a. Ulrich gibt seine Emailadresse mit Hilfe einer Tastatur auf dem Bildschirm ein.
- b. Ulrich legt seine Visitenkarte auf den Tisch. Diese wird automatisch vom Surface erkannt und die Project Notes werden an die auf der Karte aufgeführte Emailadresse verschickt.
- c. Ulrich bekommt anfangs von der Sekretärin ein getaggtes Objekt, welches zu seiner Identifizierung dient. Legt er dieses auf Surface Tisch, so werden die ausgewählten Project Notes zusammen mit seiner Identifikation an den Empfang übermittelt. Da man sich zu Beginn beim Empfang der Zühlke Engineering AG ausweisen muss, kann die Emailadresse von Ulrich ausgelesen und die Project Notes verschickt werden.

Nach der Besprechung begibt sich Ulrich zurück ins Büro der Beispielfirma. Dort angekommen ruft er seine Emails ab. Sogleich entdeckt er das Email mit den Project Notes, welches er sich von der Zühlke Engineering AG aus geschickt hat. Ulrich druckt sich zwei, die ihm besonders interessant erscheinen, aus und liest diese nun in aller Ruhe.

V.3.5.2 Persona Erik Entscheider

Erik Entscheider	
	Kurzprofil Abteilungsleiter 54 Jahre Grundlegendes technisches Verständnis und gute Computerkenntnisse
Arbeitskontext (Lärm, Unterbrüche, Regeln)	Bei der Arbeitsumgebung handelt es sich um die Eingangshalle der Zühlke Engineering AG. Falls Erik Entscheider dort warten muss, kann er sich Zeit nehmen, um sich einen Überblick über die vorangegangenen Projekte der Firma zu verschaffen. Unterbrechungen entstehen dadurch, dass er von einem Mitarbeiter abgeholt wird oder eine Unterhaltung mit jemandem beginnt, der ebenfalls wartet.
Persönlichkeit & Vorlieben	Erik verschafft sich immer gerne einen Überblick über die Firmen, mit welchen er zusammenarbeiten möchte. Daher ist es für ihn sehr wichtig herauszufinden, wie viel Fachwissen diese Firmen in gewissen Bereichen mitbringen. Zudem sind erfolgreich abgeschlossene Projekte mit namhaften Firmen für ihn immer ein Zeichen von Kompetenz.
Vorkenntnisse & Lernen (Computer, Domain)	Erik ist seit vielen Jahren Abteilungsleiter bei der Beispelfirma. Er klärt für verschiedene Projekte ab, inwiefern eine Zusammenarbeit oder Auslagerung der Arbeit Sinn macht. Durch seinen Beruf ist er ebenfalls technisch sehr bewandert und interessiert.
Eigenschaften / Verhaltensvariablen	<ul style="list-style-type: none"> • Technische Expertise • Smartphone-Kenntnisse
Ziele	<ul style="list-style-type: none"> • Zeit überbrücken • Sich über die Zühlke Engineering AG informieren

V.3.5.2.1 Ist-Szenario-1

Erik Entscheider sucht einen Partner für eine Software Lösung. Neben anderen Bewerbern würde sich auch die Zühlke Engineering AG bestens dazu eignen. Aus diesem Grund wurde Erik auf ein erstes Treffen eingeladen. Er trifft etwas verfrüh bei der Firma ein und wird daher gebeten, noch einen Moment zu warten. Erik nutzt diese Zeit, um sich weitere Informationen über Zühlke Engineering AG anzueignen. Nun ist es für Erik immer äusserst interessant zu sehen, mit welchen Firmen schon eine Zusammenarbeit stattgefunden hat oder wie umfänglich das Wissen einer Firma in bestimmten Bereichen ist. Daher überfliegt er die ausgestellten Project Notes, um nach solchen Informationen zu suchen. Dabei entdeckt er einige interessante Projekte mit namhaften Firmen und auch solchen in ihm interessant erscheinenden Themenbereichen.

Kurze Zeit später wird Erik von seinem Gesprächspartner in der Eingangshalle abgeholt. Damit er die Project Notes nach seinem Gespräch bei der Zühlke Engineering AG noch genauer durchlesen kann, nimmt er je eine Kopie mit.

V.3.5.2.2 Soll-Szenario-1

Erik Entscheider sucht einen Partner für eine Software Lösung. Neben anderen Bewerbern würde sich auch die Zühlke Engineering AG bestens dazu eignen. Aus diesem Grund wurde Erik auf ein erstes Treffen eingeladen. Er trifft etwas verfrüh bei der Firma ein und wird daher gebeten noch, einen Moment zu warten. Erik nutzt diese Zeit, um sich weiteres Wissen über Zühlke Engineering AG anzueignen. Dabei wird er von der Empfangsperson darauf hingewiesen, dass er Informationen zu abgeschlossenen Projekten über den neu erworbenen Surface Tisch einsehen kann. Gespannt setzt sich Erik vor den Tisch. Anfänglich verschafft er sich einen groben

Überblick über die verschiedenen Themenbereiche, in denen die Zühlke Engineering AG tätig ist. Danach sieht er sich die verschiedenen Firmen an und stösst so auf einige sehr spannende Projekte.

Gerade in diesem Augenblick wird Erik von seinem Gesprächspartner in der Eingangshalle abgeholt.

V.3.5.2.2.1 Alternative A

Erik entdeckte aber bereits zuvor, dass die Project Notes auch ausgedruckt werden können. Dies will er sich zu Nutzen machen und druckt die Project Notes, die er bereits vorsorglich zum Drucken markiert hat, aus.

V.3.5.2.2.2 Alternative B

Als Erik zuvor eine Project Note zum Drucken markieren wollte, wurde er aufgefordert, seinen Besucher-Badge auf den Tisch zu legen. Nun entfernt er seinen Badge wieder vom Tisch. Er ist positiv überrascht, als ihm vom Tisch mitgeteilt wird, dass seine ausgewählten Project Notes für ihn ausgedruckt werden und er sie beim Verlassen der Zühlke Engineering AG mitnehmen kann.

Nach dem Meeting gibt Erik seinen Badge ab. Die freundliche Sekretärin übergibt ihm im Gegenzug seine zuvor ausgewählten Project Notes in gedruckter Form. Erik nimmt die Ausdrucke entgegen, bedankt sich herzlich und freut sich, wie gut die internen Abläufe in der Zühlke Engineering AG funktionieren.

V.3.5.3 Persona Bettina Bewerberin

Bettina Bewerberin	
	Kurzprofil Hochschulabsolventin eines Informatik-Studiums 26 Jahre Ausgeprägtes technisches Verständnis und Computerkenntnisse
Anwendungskontext	Bei der Arbeitsumgebung handelt es sich um die Eingangshalle der Zühlke Engineering AG. Falls Bettina Bewerberin dort warten muss, kann sie sich Zeit nehmen, um sich einen Überblick über die vorangegangen Projekte der Firma zu verschaffen. Unterbrechungen entstehen dadurch, dass sie von einem Mitarbeiter abgeholt wird oder eine Unterhaltung mit jemandem beginnt, der ebenfalls wartet.
Persönlichkeit & Vorlieben	Bettina sucht nach einem erfolgreichen Hochschulabschluss nun eine Stelle. Sie möchte sich über ihren vielleicht zukünftigen Arbeitsgeber ein möglichst gutes Bild machen.
Vorkenntnisse & Lernen (Computer, Domain)	Dank ihrer Ausbildung hat Bettina ein gut ausgeprägtes technisches Verständnis. Sie hat vertiefte Kenntnisse in Software Engineering, kann User Interfaces korrekt entwerfen und kennt die Programmiersprachen Java und C++.
Eigenschaften / Verhaltensvariablen	<ul style="list-style-type: none"> • Technische Expertise • Kenntnisse in Produkt- und Software-Engineering • Smartphone-Kenntnisse
Ziele	<ul style="list-style-type: none"> • Zeit überbrücken • Sich über die Zühlke Engineering AG informieren

V.3.5.3.1 *Ist-Szenario-1*

Bettina Bewerberin hat bei der Zühlke Engineering AG ein Vorstellungsgespräch. Aus diesem Grund ist sie extra etwas früher erschienen. Bettina wird nach ihrer Anmeldung beim Empfang darauf hingewiesen, dass sie sich ja in der Zwischenzeit die ausgestellten Project Notes ansehen könnte. Dies sieht Bettina als Ansporn um Weiteres über die zu erfahren. Sie verschafft sich daher kurz einen Überblick über die ausgestellten Project Notes und beginnt dann, eine einzelne etwas genauer anzuschauen. Einerseits interessiert sie sich für das Projekt selbst, aber auch für dessen technische Daten und den Projektpartner. Bettina sieht sich die Project Notes nun mehr oder weniger der Reihe nach an, vergleicht sie und stellt für sich fest, dass gewisse Tools oder technische Daten besonders häufig aufgelistet sind. Zudem ist sie überrascht, wie vielfältig die Projekte der Zühlke Engineering AG bisher waren. Nach einiger Zeit wird sie von ihrem Gesprächspartner abgeholt.

V.3.5.3.2 *Soll-Szenario-1*

Bettina Bewerberin hat bei der Zühlke Engineering AG ein Vorstellungsgespräch. Aus diesem Grund ist sie extra etwas früher erschienen. Bettina wird nach ihrer Anmeldung beim Empfang darauf hingewiesen, dass sie ja in dieser Zeit den neu erworbenen Surface 2.0 Tisch ausprobieren könnte. Mit diesem könne sie sich über die verschiedenen Projekte der Zühlke Engineering AG informieren. Bettina setzt diesen Vorschlag sogleich in die Tat um und beginnt die verschiedenen Project Notes zu durchstöbern. Dabei entdeckt sie immer wieder neue interessante Projekte und bewegt sich von einer Project Note zur nächsten. Sie beschränkt sich dabei nicht auf ein bestimmtes Thema. Nach einiger Zeit wird sie von ihrem Gesprächspartner abgeholt.

Project Flip 20

Studienarbeit

V.4 Anforderungen

V.4.1 Dokumentinformationen

Datum	Version	Änderung	Autor
26.10.2011	1.0	Erste Version des Dokuments	cheidt
28.10.2011	1.2	Ergänzungen	dreichl
02.11.2011	1.3	Review	dreichl
15.11.2011	1.4	Review	lelmer
25.11.2011	1.5	Review	dreichl
28.11.2011	1.6	Kleine Korrekturen	dreichl
11.12.2011	1.7	Funktionale Anforderungen, NF Anforderungen	lelmer
12.12.2011	1.8	Korrekturen	dreichl
16.12.2011	1.9	Tools angepasst	lelmer
22.12.2011	1.10	Korrekturen	cheidt

V.4.2 Tools

Zur Durchführung des Projektes und Entwicklung der Software wurden die nachfolgend aufgelisteten Werkzeuge verwendet.

Tool	Version	Link
Windows 7	SP1	http://windows.microsoft.com/de-CH/windows/home
Tortoise SVN	1.7.3	http://tortoisesvn.net/
Adobe Reader	X (10)	http://get.adobe.com/de/reader/
.NET	v4.0.30319 RTMRel	http://www.microsoft.com/net
Surface 2 SDK	v2.0	http://blogs.msdn.com/b/surface/archive/2011/07/12/links-to-sdk-and-resources.aspx
Visual Studio 2010	v10.0.30319.1	
Ultimate mit Power-Tools	RTMRel	
ReSharper	6.0	http://www.jetbrains.com
dotCover	1.1.1	http://www.jetbrains.com
GhostDoc	3.0	http://submain.com/download/ghostdoc/
NDepend Trial	3.9	http://www.ndepend.com/
Expression Blend	v4.0.20525.0	http://www.microsoft.com/expression/products/blend_overview.aspx
WPF Inspector	v0.9.9	http://www.wpftutorial.net/Inspector.html
Adobe Photoshop	CS4 Extended	http://www.adobe.com/de/products/photoshop.html
Microsoft Office 2010	14 Professional Plus	http://office.microsoft.com/de-ch/
Redmine	1.3.0	http://redmine.org
Visicheck	2008-Nov-11	http://www.visicheck.com

Tabelle 2 - Tools

V.4.3 Funktionale Anforderungen

Um die funktionalen Anforderungen möglichst effizient und trotzdem exakt zu definieren, wurden User Stories als Teil von Scrum verwendet. Nachfolgend sind die User Stories nach Sprint gruppiert. Die User Stories sind mit dem jeweiligen Sprint, in welchem sie umgesetzt wurden, gekennzeichnet. Nicht umgesetzte User Stories sind mit „U“ markiert. Solche, die durch die fehlende Hardware nicht umgesetzt werden konnten mit „F“. Weitere Details sind der Tabelle User Stories im VII Anhang, C zu entnehmen.

Nachfolgend eine Übersicht über die User Stories:

Legende: F -> Future, U -> Unplanned

Titel	User Story	Definition of Done	Sprint
Übersicht für PN	Als Surface Benutzer möchte ich die PN in einer Übersicht anzeigen lassen, damit ich mir einen Überblick über die verschiedenen PN verschaffen kann.	Alle PNs werden in einem Gitter dargestellt.	3
Detailansicht PN	Als Surface Benutzer möchte ich eine Detailansicht der PN sehen, damit ich die einzelnen PN lesen kann und genauere Infos erhalte.	Eine bestimmte PN wird in einem separaten Container dargestellt, wobei die XPS Datei geöffnet sein soll.	3
Navigation "Übersicht -> Detail"	Als Surface Benutzer möchte ich von der Übersicht über alle PN in eine Einzelansicht wechseln, damit ich die einzelnen PN lesen kann und genauere Infos erhalte.	Durch das Anklicken einer PN in der Übersicht wird die Detailansicht mit genau dieser PN geöffnet.	3
Navigation "Detail -> Übersicht"	Als Surface Benutzer möchte ich von der Detailansicht zur Übersicht navigieren, damit ich mir einen Überblick über die aktuelle Detailansicht verschaffen kann.	Durch das Anklicken eines zurück-Buttons wird die Detailansicht geschlossen und die Übersicht wird wieder angezeigt.	3
Darstellung der verkleinerten PN mit Bild	Als Surface Benutzer möchte ich die PN mithilfe eines Bildes unterscheiden können, wenn sie verkleinert sind, um einfacher eine Auswahl zu treffen.	Die PNs werden in einem Gitter mit je einem Bild dargestellt.	3
Navigation "Detailansicht -> Detailansicht"	Als Surface Benutzer möchte ich bei der Detailansicht einer PN zur nächsten oder vorherigen PN wechseln können, damit ich alle PN nacheinander anschauen kann.	In der Detailansicht kann auf einen Button "nach rechts" oder "nach links" geklickt werden, wobei die aktuelle PN mit der rechten bzw. linken PN ausgetauscht wird.	3
Tags zu PN angezeigt	Als Surface Benutzer möchte ich zu einer PN zusätzliche Begriffe anzeigen können, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Tags, die zu einer PN gehören, werden in der Detailansicht zur PN angezeigt.	4
Tags aggregiert	Als Surface Benutzer möchte ich bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Die Tags werden aggregiert angezeigt.	4

Aggregierte Tags anpassbar	Als Surface Administrator möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.	Config File kann von einem Admin bearbeitet werden.	4
Filter in Übersicht setzen	Als Surface Benutzer möchte ich einen Filter in der Übersicht setzen können, damit die angezeigten PN eingeschränkt werden.	In der Übersicht kann ein Filter gesetzt werden, sodass alle PN nach dem gesetzten Filter gefiltert werden. Die tatsächliche Auswahl des Filters ist hier nicht inbegriffen.	4
Filterkriterium auswählen	Als Surface Benutzer möchte ich das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann.	Ein Filterkriterium kann aus einer Menge von Filterkriterien ausgewählt werden.	4
Filter in Übersicht entfernen	Als Surface Benutzer möchte ich einen Filter in der Übersicht entfernen können, damit mir mehr PN angezeigt werden.	Ein zuvor gewähltes Filterkriterium kann wieder entfernt werden, wobei die PN nicht mehr nach diesem Filterkriterium gefiltert werden.	4
Filter in Detailansicht entfernen	Als Surface Benutzer möchte ich einen Filter in der Detailansicht entfernen können, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	In der Detailansicht kann der Filter entfernt werden.	4
Externes Design festgelegt und validiert	Als Surface Benutzer möchte ich Als Entwickler möchte ich für die Design User Stories eine "Definition of Done" festlegen können, damit der Abschluss der User Stories validiert werden kann.	Das externe Design wurde im Photoshop erstellt und dem Kunden gezeigt, der damit zufrieden war.	4
Filter in Detailansicht setzen	Als Surface Benutzer möchte ich einen Filter in der Detailansicht setzen können, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	In der Detailansicht kann ein Filter gesetzt werden, wobei nach dem Setzen des Filters zur Übersicht gewechselt wird.	5
Filtern nach mehreren Kriterien	Als Surface Benutzer möchte ich nach mehreren bestimmten Kriterien filtern können, um eine begrenzte Auswahl zu erhalten.	Es können mehrere Filter gleichzeitig eingegeben werden, wobei nach allen gefiltert wird (AND).	5
Animierte Navigation "Übersicht -> Detail"	Als Surface Benutzer möchte ich durch eine Animation erkennen können, dass von der Übersicht in die Detailansicht (oder in die Gegenrichtung) gewechselt wurde, damit der Programmfluss für mich verständlicher ist.	Der abgedunkelte Hintergrund kann angeklickt werden, wodurch zur Übersicht gewechselt wird.	5
Scrollerkennung in Übersicht dargestellt	Als Surface Benutzer möchte ich an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann.	In der Übersicht existiert ein grafisches Element, welches auf mehr Project Notes hindeutet.	5
Schöne Darstellung der Tags	Als Surface Benutzer möchte ich eine ansprechende Darstellung der Begriffe, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Die Tags werden gemäss externem Design dargestellt.	5

Schöne Darstellung des Filters	Als Surface Benutzer möchte ich eine ansprechende Darstellung des Filters, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Der Filter wird gemäss dem externen Design dargestellt	5
Lesemodus PN anzeigen	Als Surface Benutzer möchte ich eine PN vergrössern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann.	Die PN kann so weit verbreitert werden, wie neben dem Filter und den Tags noch Platz ist.	6
Animation für Navigation "Detailansicht -> Detailansicht"	Als Surface Benutzer möchte ich eine Animation sehen können, wenn ich durch die PNs navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat.	Es existiert eine Animation, wenn von der einen Detailansicht in die andere Detailansicht gewechselt wird.	6
Anzahl PN bei Übersicht anzeigen	Als Surface Benutzer möchte ich die Anzahl PN in der Übersicht sehen können, damit ich weiss, wie gross die Liste ist.	Oberhalb des Filters wird dargestellt, wie viele von der maximalen Anzahl an Project Notes aktuell in der Übersicht angezeigt werden.	6
Eastereggs	Als Entwickler oder Zühlke Engineering AG Mitarbeiter möchte ich ein Eastereggs ausführen können, damit ich Spass haben und Kunden beeindrucken kann.	Info View für Beteiligte Parteien & Personen erscheint beim Klicken des Info Buttons.	6
Animation Filter -> Übersicht	Als Surface Benutzer möchte ich eine Animation sehen, wenn die Ansicht geschlossen und der Filter gesetzt wird, damit ich verstehe, dass dieser nun gesetzt ist und sich die Übersicht dementsprechend angepasst hat.	Wenn ein Filterkriterium gesetzt wird, so wird es eine Animation ausgelöst.	6
Animation Filterkriterium	Als Surface Benutzer möchte ich, dass sich die Unterkategorie einblendet, wenn ich eine Oberkategorie auswähle, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.		F
PN ausgedruckt	Als Surface Benutzer möchte ich eine PN ausdrucken können, um sie mitzunehmen und später genau lesen zu können.		F
Badge erkennen	Als Surface Benutzer möchte ich einen Besucher Badge auf den Surface legen können, welcher dann automatisch erkannt wird und mir entsprechende Informationen anzeigt.		F
PN verschickt	Als Surface Benutzer möchte ich eine PN an eine Emailadresse schicken können, um sie später genau lesen zu können.		F
Animation für aktionslose Beschriftung	Als Surface Benutzer möchte ich beim Drücken einer aktionslosen Beschriftung Feedback erhalten, damit ich erkenne, dass keine Aktion hinter der Beschriftung steht.		U
Volltextsuche	Als Surface Benutzer möchte ich mithilfe einer Volltextsuche PNs filtern können, damit ich nur diese		U

	anzeigen lassen kann, die mich wirklich interessieren.	
Demomodus erstellt	Als Surface Benutzer möchte ich durch einen Demomodus auf die Applikation aufmerksam gemacht werden, damit ich animiert werde, den Surface zu bedienen.	U
Demomodus erkennen und verlassen	Als Surface Benutzer möchte ich den Demo Modus erkennen, damit ich diesen verlassen kann, um selbst navigieren zu können.	U
Animation für Lesemodus	Als Surface Benutzer möchte ich eine Animation sehen, wenn ich in den Lesemodus wechsle, damit mir die Applikation mehr Spass macht.	U
Animation Übersicht -> Filter	Als Surface Benutzer möchte ich eine Animation sehen können, wenn die Kriterienliste für das Filtern geöffnet wird, damit ich den Filtermodus besser erkenne.	U

Tabelle 3 - User Stories

V.4.4 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen lassen sich zum Teil aus den User Stories ableiten. Einige Anforderungen, wie z.B. die Wartbarkeit, können jedoch nicht daraus abgeleitet werden. Deshalb ist es notwendig, diese in diesem Dokument festzuhalten.

V.4.4.1 Funktionalität

V.4.4.1.1 Angemessenheit

Die Zeit, während der ein Anwender die Applikation benutzt, ist kurz. Die Software soll für ihren Zweck der Bereitstellung von Informationen über bisherige Projekte der Zühlke Engineering AG und auch zur Unterhaltung während einer kurzen Zeitdauer gut geeignet sein. Darum soll sie eine Funktionalität mit einer begrenzten Anzahl an Funktionen zur Verfügung stellen (browsen, filtern, lesen). Die Angemessenheit wird durch einen Usability Tests gemessen und verifiziert.

V.4.4.2 Zuverlässigkeit

V.4.4.2.1 Fehlertoleranz

Das System soll so aufgebaut sein, dass Fehleingaben nicht möglich sind. Wenn nach mehreren Kriterien gefiltert wird und keine Project Note diesen Kriterien entspricht, so soll der Benutzer auf diesen Umstand aufmerksam gemacht werden. Es muss zudem möglich sein, zur vorherigen Situation zurückzukehren. Dies wird durch heuristische Evaluation sichergestellt.

V.4.4.3 Benutzbarkeit

V.4.4.3.1 Verständlichkeit & Erlernbarkeit

Durch die begrenzte Zeit, die der Nutzer vor der Applikation verbringt, muss diese einfach benutzbar sein. Die Bedienung soll ohne Benutzerhandbuch oder anderweitige Hilfe möglich sein. Viel eher soll der Nutzer mit jeder Berührung des Tisches etwas Neues lernen und dieses sogleich anwenden können.

Ein technisch affiner Benutzer soll während des Usability-Tests ohne zusätzliche Hilfe die Bedienung der Software erlernen können.

V.4.4.3.2 Bedienbarkeit

Die Bedienung soll einfach und intuitiv sein. Die einzelnen Komponenten sollen einfach erreichbar sein und über eine genügend grosse Fläche verfügen, damit sie anklickbar sind.

In einem Usability Test werden verschiedene Szenarien vorgegeben, die dann von Testpersonen ohne Erfahrung mit der Applikation und ohne ein Benutzerhandbuch durchgeführt werden. Es müssen alle Szenarien erreicht werden können, ohne dass die Testpersonen Hilfe beanspruchen müssen.

V.4.4.3.3 Attraktivität

Durch das externe Design soll sofort ersichtlich sein, dass es sich um eine Applikation der Zühlke Engineering AG handelt. Dies wird durch das Übernehmen des Corporate Designs sichergestellt.

Durch eine Fernwirkung, beispielsweise durch einen Demomodus, soll der Nutzer auf die Applikation aufmerksam gemacht und angelockt werden. Eine Nahwirkung wird durch verschiedene Animationen bei der Bedienung gewährleistet. Diese sollen auf den Nutzer ansprechend wirken und ihn dadurch dazu verführen, sich mit der Anwendung auseinanderzusetzen. Gerade bei kurzer Nutzung ist der erste Eindruck entscheidend.

Das Visuelle Design wurde von der Zühlke Engineering AG validiert und akzeptiert.

V.4.4.4 Effizienz

V.4.4.4.1 Zeitverhalten

Durch den begrenzten Zeitrahmen soll es einem technisch affinen Benutzer innerhalb von drei Minuten möglich sein, ein gewünschtes Projekt oder einen gewünschten Themenbereich zu finden um sich mit diesem auseinandersetzen zu können.

V.4.4.5 Änderbarkeit & Wartbarkeit

Die Software soll für die Zühlke Engineering AG möglichst einfach änderbar sein. Es wird deshalb auf die Codequalität geachtet, wobei ReSharper genutzt wird, um die Qualität zu prüfen (orange Markierungen auf der rechten Seite des Editors). Es sollen im Schnitt maximal drei solche orangen Markierungen pro C# Datei (.cs) erscheinen.

Zusätzlich sollen die Code Metriken beachtet werden. Ziel ist es, einen „Maintainability Index“ [microsoft11.2] von mindestens 50% zu erreichen, und zwar auf Ebene Projekt. Ausnahmen können mündlich begründet werden, speziell bei den Testprojekten.

Um die Software möglichst einfach warten zu können, sollen die Kategorien dynamisch anpassbar sein. Dafür soll ein Tab-getrenntes Textdokument zur Konfiguration der anzuzeigenden Tags bereitgestellt werden. Das Textfile soll in ein Tabellenkalkulationsprogramm (z.B. Excel) importiert und nach den Änderungen wieder in ein Textdokument abgespeichert werden können.

V.4.5 Design Constraints

Vorgegeben ist als Hardware der Surface 2 Tisch der Microsoft Corporation. Durch die Verwendung dieser Hardware ist auch die Software-Plattform automatisch auf .NET beschränkt.

Die Zühlke Engineering AG wünscht sich, dass möglichst wenig Aufwand für den Unterhalt des Tisches nötig sein wird. Daher muss mit Project Notes im Originalformat PDF gearbeitet werden.

Die Kategorien und deren zugehörige Begriffe, die zu einer Project Note gehören, sind ebenfalls über die Metadaten vordefiniert. Diese können von der Applikation aggregiert oder gegebenenfalls ignoriert werden.

Das Corporate Design gibt zudem die Richtlinien für das externe Design vor.

V.4.6 Zugänglichkeit (Accessibility)

Der Surface Tisch ist ein Multitouch Gerät und wird daher mit den Händen bedient. Die Applikation ist deshalb nicht darauf ausgerichtet, mit einer Tastatur oder sonstigen Eingabegeräten benutzbar zu sein.

Eines der Hauptmerkmale von Project Flip 2.0 ist das Lesen einer Project Note. Durch diese zentrale Rolle kann die Applikation nicht von Menschen mit einer Sehbehinderung verwendet werden. Zwar könnte eine Project Note mit einer Vorlese-Anwendung vorgetragen werden, aber die Navigation kann damit nicht gelöst werden. Zudem würde dies andere Wartende oder das Empfangspersonal ablenken.

Personen mit partiell-funktionalen Sehbehinderungen wie Farbenblindheit soll es hingegen möglich sein, die Applikation zu verwenden.

Als Ausgangslage wurde der Startscreen, welcher im externen Design definiert wurde (siehe V.5.3.4 Externes Design), verwendet.



Abbildung 4 - Startscreen, original

Dieser wurde nachfolgend mit Vischeck² geprüft. Die Webseite bietet ein Plugin an, welches drei Arten von Farbenblindheit simuliert.

² Quelle: <http://www.vischeck.com/>

1. Deutanopie oder Grünblindheit

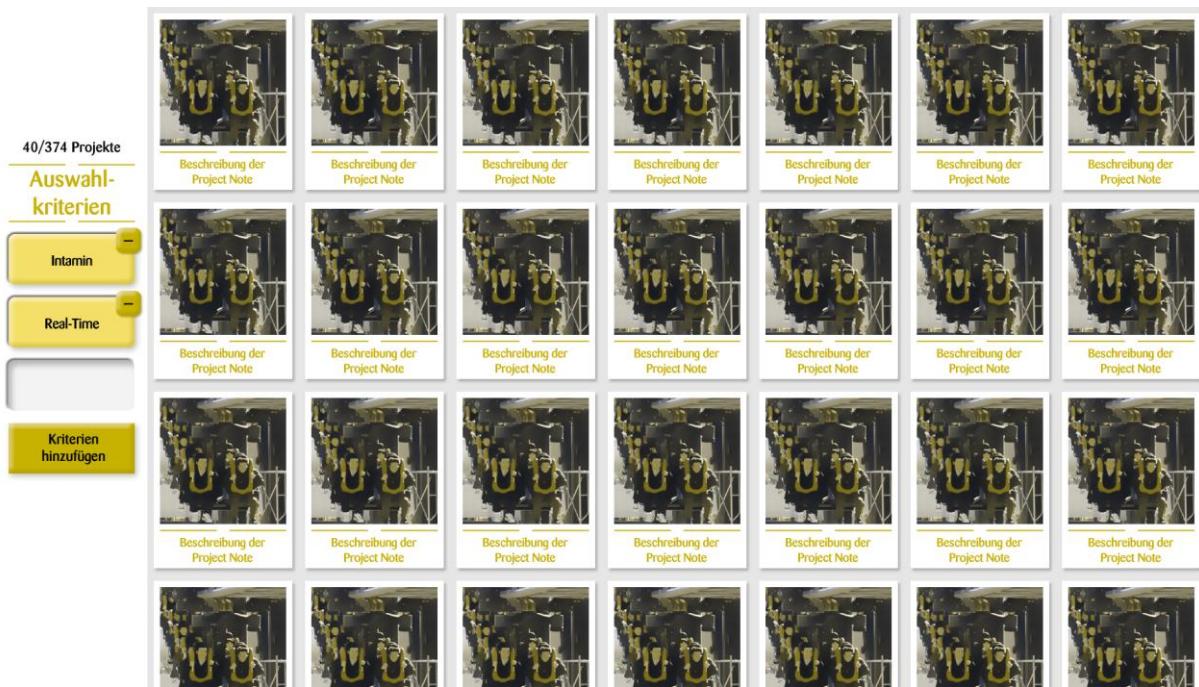


Abbildung 5 - Startscreen, Deutanopie

2. Protanopie oder Rotblindheit. Diese Form ist auch bekannt als Rotgrünblindheit erster Form

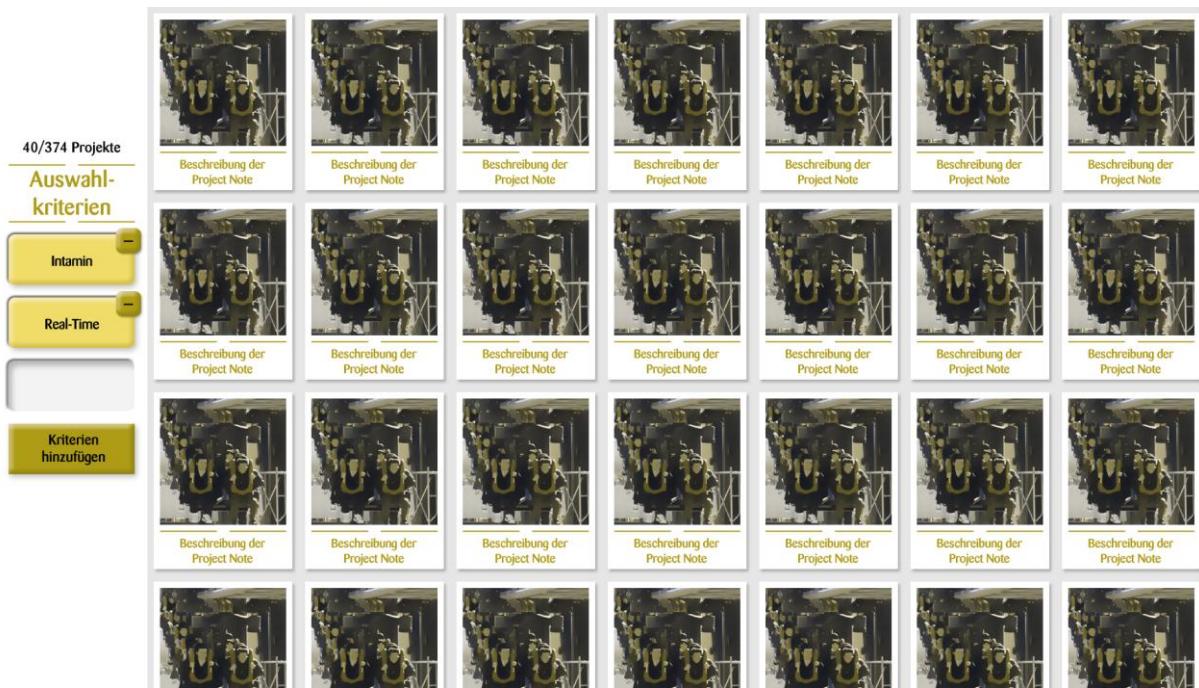


Abbildung 6 - Startscreen, Protanopie

3. Tritanopie oder Blaublindheit.



Abbildung 7 - Startscreen, Tritanopie

Wie aus den Abbildungen ersichtlich ist, ist der Kontrast immer noch genug hoch, so dass die Texte lesbar und die einzelnen Elemente weiterhin erkennbar sind.

Project Flip 20

Studienarbeit

V.5 Domain Analyse

V.5.1 Dokumentinformationen

Datum	Version	Änderung	Autor
07.10.2011	1.0	Erste Version des Dokuments	cheidt
16.10.2011	1.1	Review, Überarbeitung, weitere Punkte	lelmer
16.10.2011	1.2	Eintragung Papier-Prototypen & Korrekturen	cheidt
17.10.2011	1.3	Review, grammatischen Korrekturen	dreichl
17.10.2011	1.4	Domain Modell eingefügt	lelmer
23.10.2011	1.5	Eintragung Guidelines	cheidt
24.10.2011	1.6	Korrekturen aus Sitzung	cheidt
24.10.2011	1.7	Domain Analyse ergänzt	lelmer
24.10.2011	1.8	Review Guidelines	dreichl
31.10.2011	2.0	Korrektur Prozessmodell	dreichl
01.11.2011	2.1	Review Prozessmodell	cheidt
04.11.2011	2.2	Pivot Viewer	cheidt
04.11.2011	2.3	Review Pivot Viewer	dreichl
14.11.2011	2.4	Externes Design	cheidt
14.11.2011	2.5	Review externes Design	lelmer
14.11.2011	2.6	Korrekturen Externes Design	dreichl
28.11.2011	2.7	Objekt Diagramm	cheidt
28.11.2011	2.8	Review	dreichl

V.5.2 Daten

In den Metadaten einer Project Note sind verschiedene Angaben gespeichert. Darin enthalten sind beispielsweise Informationen zum Kunden, mit welchem das Projekt realisiert wurde. Auch die Technologien, mit denen gearbeitet wurde oder der Fokus des Projektes sind in den Informationen enthalten. Diese Angaben werden verwendet, um die Project Notes zu filtern.

V.5.2.1 Prozessdiagramm

Bis eine Project Note im Eingangsbereich der Zühlke Engineering AG ausgestellt werden kann, müssen folgende Schritte durchlaufen werden:



Abbildung 8 - Prozessdiagramm

V.5.2.2 Domain Model

Eine Project Note kann grundsätzlich mehrere Begriffe derselben Kategorie enthalten. Folgendes Domain Model zeigt die Relationen:

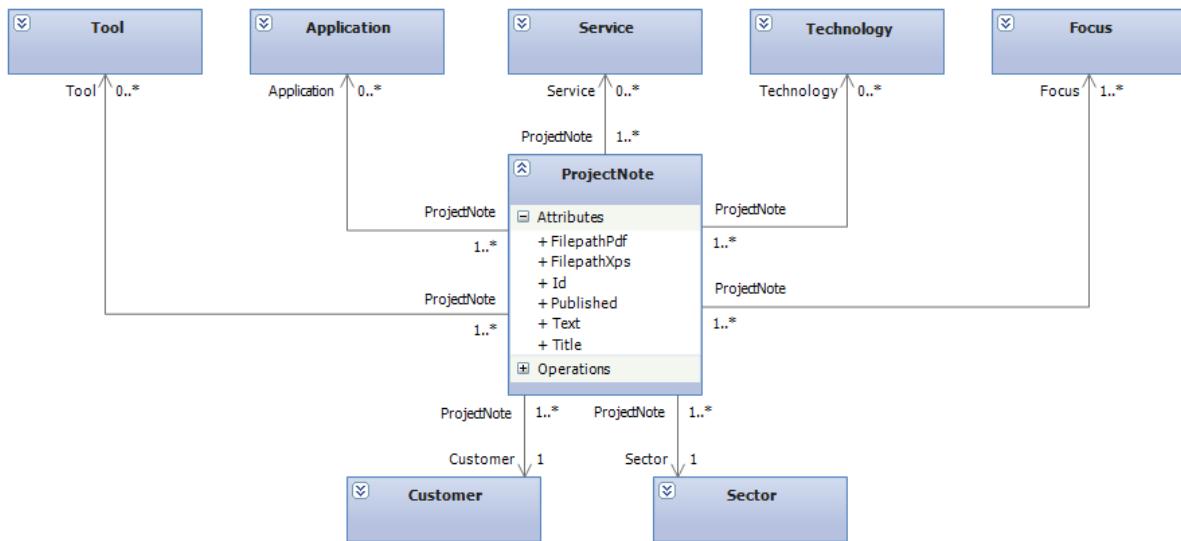


Abbildung 9 - Domain Model Daten

Ausser der Klasse ProjectNote beinhaltet jede Klasse ein Attribut Name (String), wie z.B. „Swisscom“ für ein „Customer“ Objekt.

Zum besseren Verständnis illustriert das folgende Objektdiagramm ein Beispiel einer Project Note.

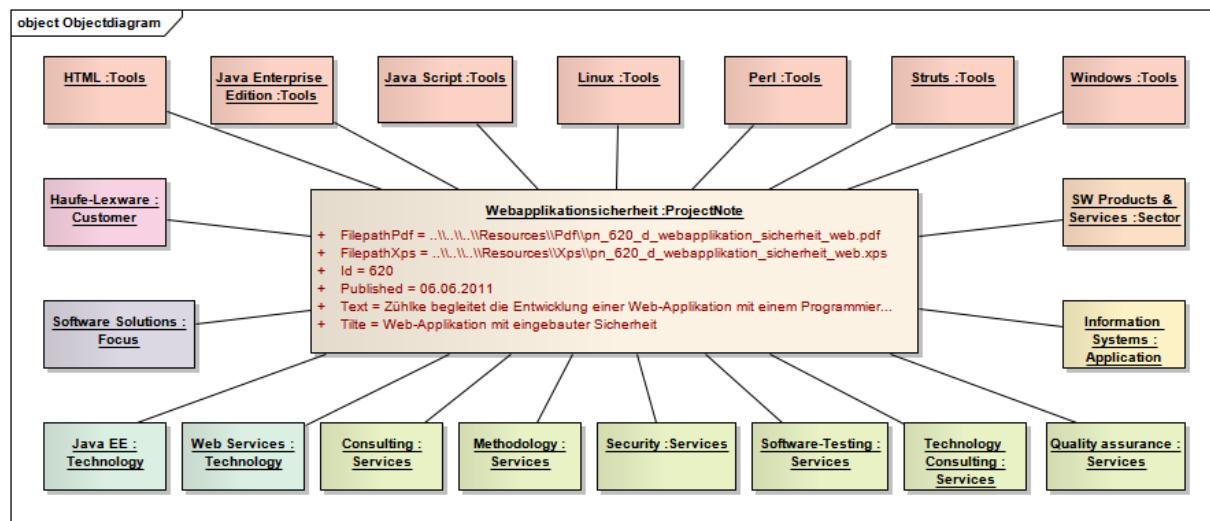


Abbildung 10 - Objekt Diagramm Project Note

V.5.2.3 Kategorien und Zahlen

Die Kategorien sind die folgenden:

Bereich	Anzahl Begriffe	Minimale Anzahl Project Notes	Maximale Anzahl Project Notes	Beispieldaten
Sektoren	15	4	52	Machinery & Plants
Kunden (Firmen)	207	1	19	Swisscom
Fokusse	7	1	223	Software Solutions
Dienstleistungen	55	1	153	Product Innovation
Technologien	18	1	35	C#
Applikationen	15	1	48	Information Systems
Werkzeuge	35	1	46	Palm OS

Tabelle 4 - Gegebene Daten im Zusammenhang mit den Project Notes

Pro Bereich (Kategorie) existieren 7 bis 207 verschiedene Begriffe, welchen zwischen 1 bis 223 verschiedene Project Notes zugeordnet sind.

V.5.2.4 Aggregation der Kategorien und Begriffe

In einer Besprechung mit der Zühlke Engineering AG wurde beschlossen, die Anzahl der Kategorien und auch der Begriffe zu begrenzen. Eine Aggregation geschieht, indem z.B. Java EE, Java ME, J2EE unter dem Begriff Java zusammengefasst werden. Falls eine Aggregation nicht möglich sein sollte, können einige wenige Begriffe auch vollständig gestrichen werden.

Um die Wartbarkeit der Applikation zu gewährleisten, werden die Zuordnungen in einer Konfigurationsdatei gespeichert. So muss bei einer Änderung der Zuordnung der Code nicht erneut kompiliert werden.

V.5.3 Graphical User Interface (GUI)

V.5.3.1 Creative Workshop

Um möglichst viele Varianten zur Darstellung der Project Notes (PN) zu finden, wurde am 07.10.2011 ein Creative Workshop mit allen Teammitgliedern durchgeführt. Vor dem Workshop liess sich das Team durch [shen06] inspirieren. Nachfolgend sind die Ergebnisse des Workshops kurz zusammengefasst.

V.5.3.1.1 Allgemeiner Ablauf des Programms

Sobald das Gerät gestartet wird, ist es im Zustand „Start“. In diesem Zustand wird die auf dem Surface angezeigte Project Note alle 30 Sekunden automatisch ausgewechselt (im Weiteren als „Demo Modus“ beschrieben). Beginnt der Benutzer die Applikation zu verwenden, wird in einen zweiten Zustand gewechselt. Nun wird die PN nicht mehr automatisch gewechselt, damit der Kunde selbst Browsen und auch einen Filter setzen kann. Geschieht während 5 Minuten keine Interaktion mit dem Gerät, so erfolgt der Wechsel in den Demo Modus.

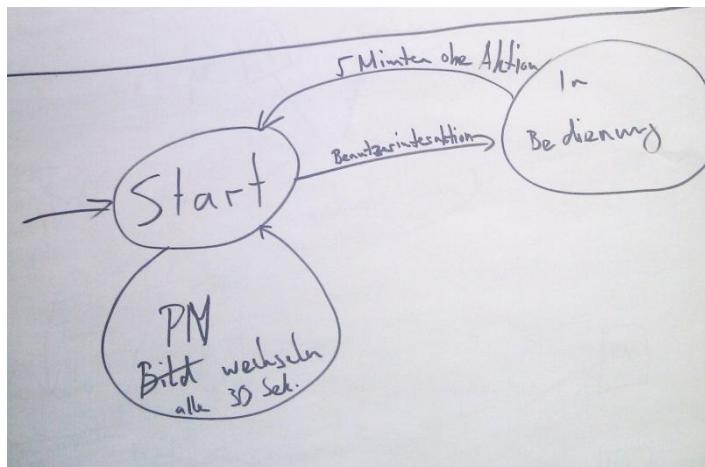


Abbildung 11 - Allgemeiner Ablauf

V.5.3.1.2 Tree View

Zu Beginn zeigt die Applikation einen gewurzelten Baum (siehe Abbildung 12 - Übersicht Tree View). Der Wurzelknoten, welcher in den nachfolgenden Abbildungen als kleines Haus dargestellt ist, hat mehrere Kindknoten. Diese stellen die Kategorien dar, unter welchen die in den Project Notes enthaltenen Angaben gespeichert sind (siehe dazu Kapitel V.5.2 Daten).

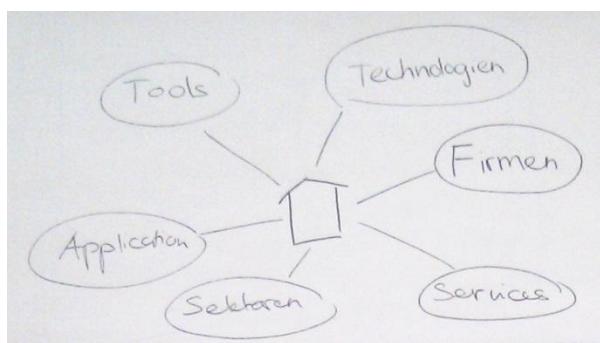


Abbildung 12 - Übersicht Tree View

Sobald ein Kindknoten angetippt wird, bewegt sich dieser in die Mitte des Bildschirms und seine Unterknoten werden dargestellt (siehe Abbildung 13 - Tree View Stufe 1). Die restlichen Knoten ordnen sich neu um den Knoten in der Mitte an und verkleinern sich gegebenenfalls.

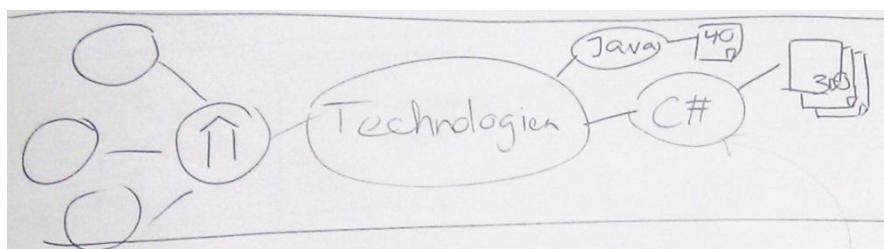


Abbildung 13 - Tree View Stufe 1

Von jedem Knoten aus kann so lange tiefer in den Baum navigiert werden, bis ein äusserster Knoten (in Abbildung 13 - Tree View Stufe 1 entsprechen die Knoten Java und C# je einem äussersten Knoten) erreicht ist. Der dabei durchlaufene Pfad ist stets sichtbar.

Wird ein äusserer Knoten angewählt, so werden die diesem Knoten zugehörigen Project Notes in minimierter Form dargestellt (siehe Abbildung 14 - Tree View Stufe 2).

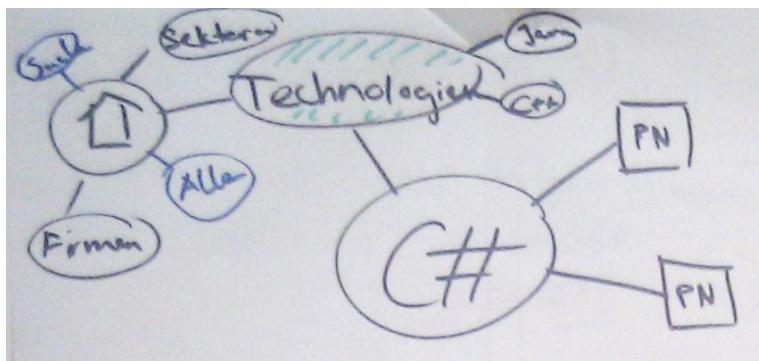


Abbildung 14 - Tree View Stufe 2

Bei der Auswahl einer im Kleinformat dargestellten Project Note öffnet sich diese. Dabei verkleinern oder verschwinden die anderen Elemente, damit für die Darstellung der Project Note genügend Platz besteht.

Neben der gross dargestellten Project Note werden Knoten angezeigt, die die zusätzlich vorhandenen Angaben über die Project Note auflisten. Dies soll dem Nutzer einerseits helfen, sich über den Inhalt der PN einen Überblick zu verschaffen. Andererseits soll es den Benutzer dazu animieren, Stichworte oder Kriterien zu entdecken, die ihn interessieren und diesen zu folgen.

Für diese Anzeige wurden die nachfolgenden Varianten A und B erarbeitet.

V.5.3.1.2.1 Variante A

Variante A1

Die für die aktuelle Project Note relevanten Begriffe befinden sich in einer Art „Wolke“, um sich von den oben dargestellten Knoten, welche zur Übersicht dienen, abzuheben. Sobald einer der Begriffe in der Wolke ausgewählt wird, öffnet sich eine neue Ansicht mit einem gewurzelten Baum (analog Abbildung 14 - Tree View Stufe 2). In dieser befindet sich das eben ausgewählte Objekt im Zentrum, welches zusammen mit den daran hängenden Knoten dargestellt wird. Die Darstellung des Baumes geschieht mittels einer Animation.

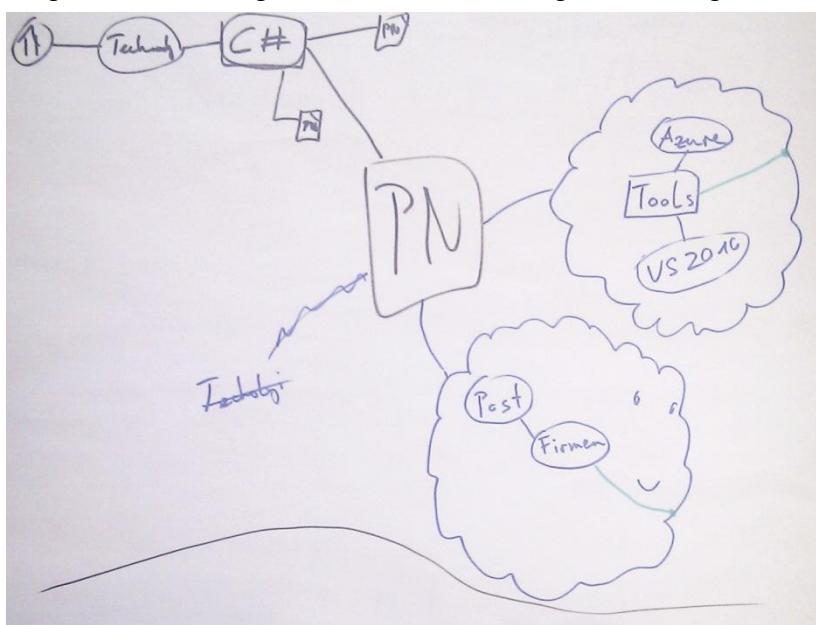


Abbildung 15 - Tree View, Variante A1

Variante A2

Die zweite Variante ist von der Funktion und der Optik her etwas anders. Wird ein äusserer Knoten angetippt, werden alle diesem Knoten angehängten Project Notes in einer Detailansicht dargestellt (für Erklärungen zur Detailansicht siehe Unterkapitel V.5.3.1.6 Detailansichten). Wieder dient der bei der Auswahl der Knoten durchlaufene Pfad zur Übersicht (am oberen Rand dargestellt). Die verschiedenen Begriffe der in der Detailsicht

aktuell dargestellten Project Note werden rechts neben der Ansicht aufgelistet. Sobald einer dieser Begriffe ausgewählt wird, wechseln oben in der Navigation die Knoten und die Detailansicht passt sich entsprechend an.

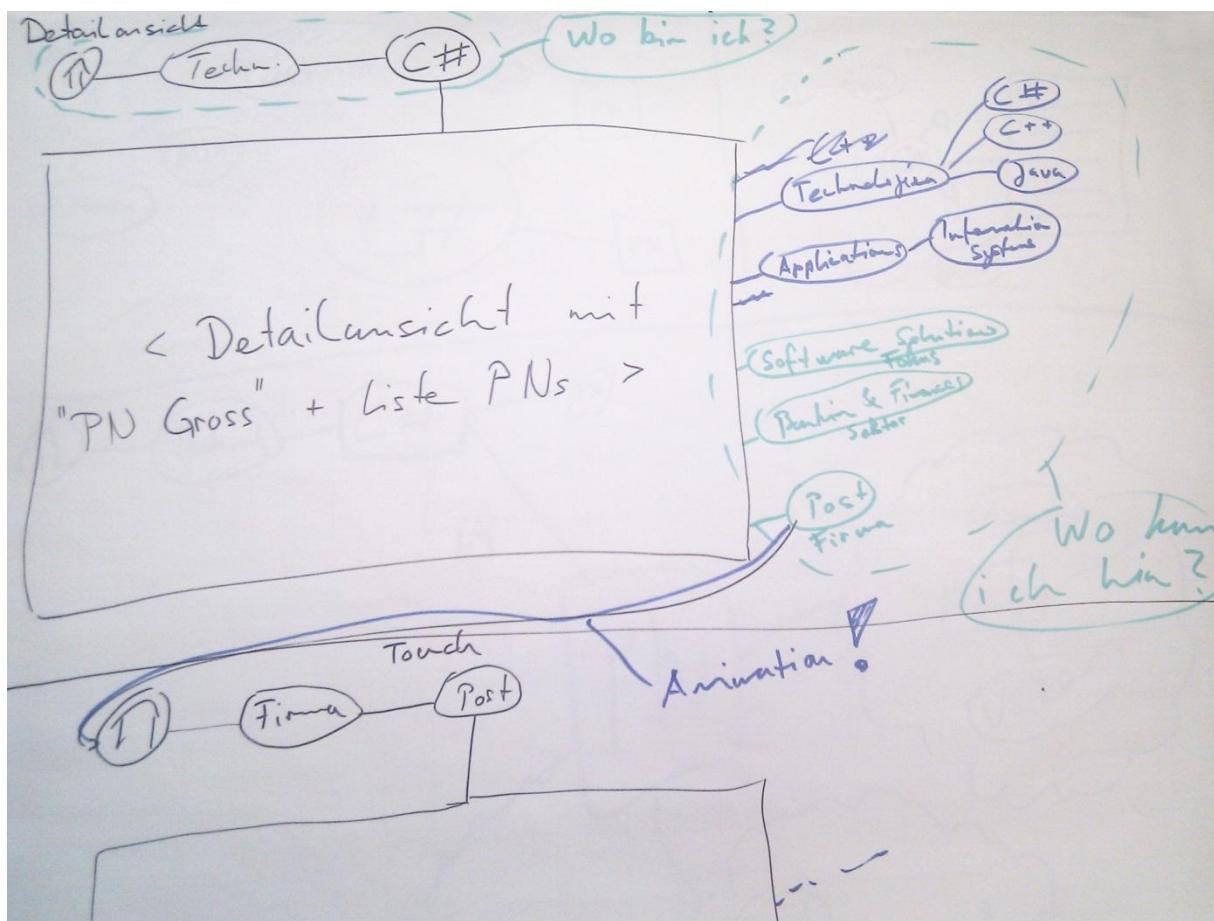


Abbildung 16 - Tree View, Variante A2

Fazit Variante A

Bei beiden Varianten ergeben sich folgende Probleme:

Verschiedenen Elemente werden doppelt aufgelistet. Als Beispiel siehe Abbildung 15 - Tree View, Variante A1: Der Knoten Technologien ist zweimal aufgelistet. Das erste Mal bei der Hauptnavigation zur Project Note, das zweite Mal in der Wolke mit den Informationen zu der jeweiligen Project Note.

V.5.3.1.2.2 Variante B

Die ausgewählte Project Note befindet sich in der Mitte des Bildschirmes, die verschiedenen Kategorien ordnen sich immer um das gerade ausgewählte Element an. Die verschiedenen Begriffe, welche sich auf die aktuell dargestellte Project Note beziehen, werden am jeweiligen Kategorienknoten angehängt und so dargestellt. An den äussersten Knoten werden die zugehörigen Project Notes in minimierter Form angezeigt.

Durch diese Art der Darstellung der Knoten entfällt die doppelte Auflistung der Elemente, welche in Variante A (Unterkapitel V.5.3.1.2.1 Variante A) das Problem ist.

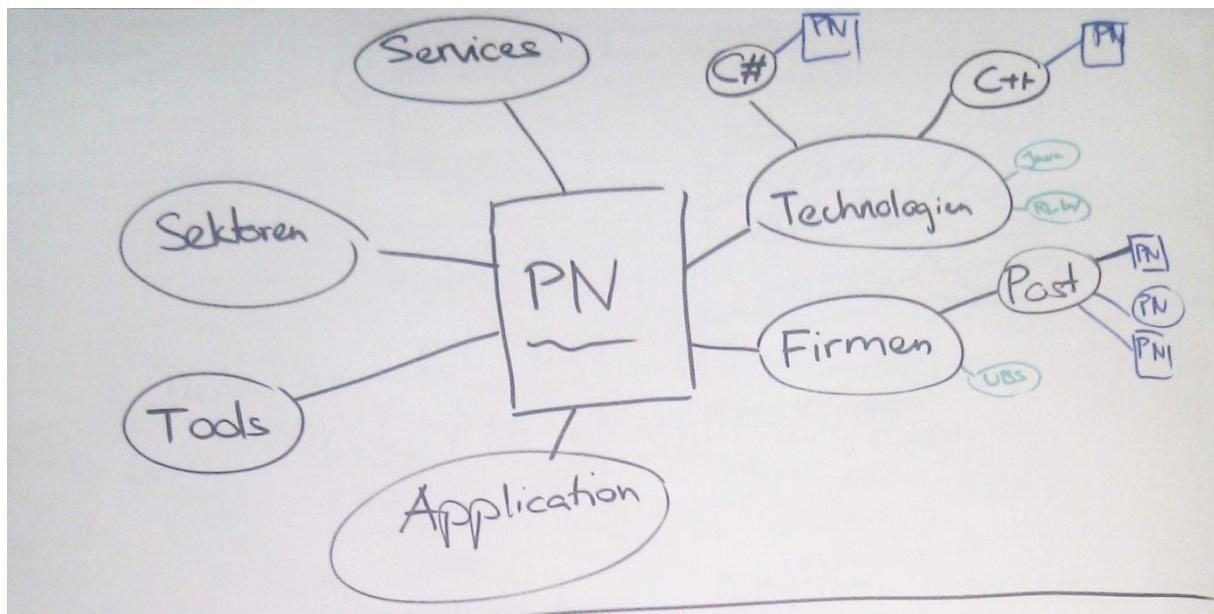


Abbildung 17 - Tree View, Variante B

Bei dieser Variante ergeben sich folgende Probleme:

Eine Project Note besitzt Angaben zu mehreren Kategorien. Das Projekt einer der am Knoten „Post“ angehängten Project Notes ist beispielsweise mit der Technologie C++ erstellt worden. Nun wird diese bestimmte PN zweimal dargestellt, einmal am Knoten „Post“ und das zweite Mal am Knoten „C++“.

Es ist nicht ersichtlich, welcher Pfad zur in der Bildschirmmitte dargestellten Project Note führte, da eine Navigationshilfe fehlt.

V.5.3.1.2.3 Fazit

Im Allgemeinen ergeben sich mit der Tree View zusätzlich die folgenden Probleme:

An gewissen Knoten sind so viele Project Notes angehängt, dass diese nicht alle gleichzeitig dargestellt werden können. Dies ist aber nicht nur auf der Höhe der Project Notes-Knoten der Fall, sondern schon bei der Auflistung der verschiedenen Firmen.

Zudem werden Project Notes mehrfach aufgelistet: Die PN eines Projektes, welches mit den Technologien C# und C++ erstellt worden ist, wird bei beiden Kategorien angezeigt.

Die Tree View wurde daher für dieses Projekt als nur teilweise geeignet eingestuft.

V.5.3.1.3 Spider View

In der Spider View werden alle Project Notes als eine Art Netz dargestellt. In der Mitte wird jeweils die aktuelle PN so dargestellt, dass der Benutzer deren Inhalt lesen kann. Wählt der Benutzer eine andere Project Note aus, so wird zu dieser navigiert, wobei die vorherige PN geschlossen und die neue PN wieder in der Bildschirmmitte geöffnet wird.

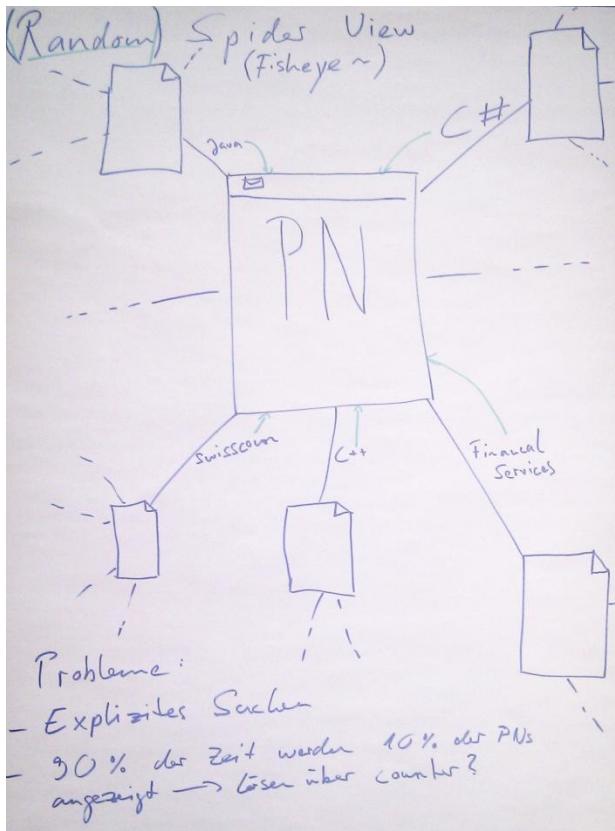


Abbildung 18 - Spider View

Der Aufbau geschieht hierbei wie bei einem Baum und enthält keine Zyklen. Damit dieser nicht immer gleich aussieht, wird der Baum zufällig aufgebaut (z.B. ein Mal pro Tag). Dies funktioniert folgendermassen:

1. Zufällige PN auswählen, diese als Startknoten zeichnen.
2. 3-5 ähnliche PN pro neu gezeichnete PN suchen, zeichnen und dann mit dieser PN verbinden. Die Verbindung wird mit einem übereinstimmendem Attribut beschrieben (z.B. C#).
3. Weiter mit Schritt 2 so lange nicht alle PN gezeichnet sind.

Wie im Unterkapitel V.5.3.1.1 Allgemeiner Ablauf des Programms beschrieben, wechseln die Project Notes im Demo Modus ca. alle 30 Sekunden. In der Spider View wird dies so realisiert, dass eine zufällige PN gewählt wird und dann von der aktuellen PN zu der neu gewählten PN navigiert wird. Schön wäre natürlich, wenn sich dazu eine Animation mit Easing Effekt verwenden liesse.

V.5.3.1.3.1 Erweiterungen

- Bei der Navigation zwischen zwei Project Notes wird mit einer Animation gearbeitet.
- Dem Benutzer werden ein Vor- und Zurück-Button zur Verfügung gestellt, damit er in einer bestimmten Reihenfolge durch die Project Notes browsen kann.
- Die aktuelle Project Note kann beliebig auf dem Bildschirm umhergezogen werden.
 - Das Netz um die herumgezogene PN herum wird dadurch verdrängt. Dies könnte z.B. mit einer Physik-Engine und „unsichtbaren Federn“ zwischen den einzelnen PN realisiert werden.
- Zoom: Der Benutzer kann frei hinein und heraus zoomen mit Gesten.
- Querverbindungen zwischen Project Note erlauben → Zyklen im Graph erlauben.

- Das Netz wird so aufgebaut, sodass durch zufälliges Browsen jede Project Note möglichst gleich häufig ausgewählt wird.
- Beim Zeichnen des Baumes wird darauf geachtet, dass Gruppierungen entstehen (Bsp. Kunde: Swisscom).
- Man kann nach gewissen PN Attributen filtern (siehe Unterkapitel V.5.3.1.4 Multi-Filter)

V.5.3.1.4 Multi-Filter

Zu Beginn wird eine Detailansicht dargestellt, in der alle PN sichtbar sind. Durch das Drücken des „Plus“ erscheint als zweiter Layer über der bisherigen Ansicht eine Tree View (siehe Unterkapitel V.5.3.1.2 Tree View), aus welcher nun Filterelemente ausgewählt werden können. Ist die Auswahl getroffen, kann die Tree View durch erneutes Drücken des „Plus“ geschlossen werden.

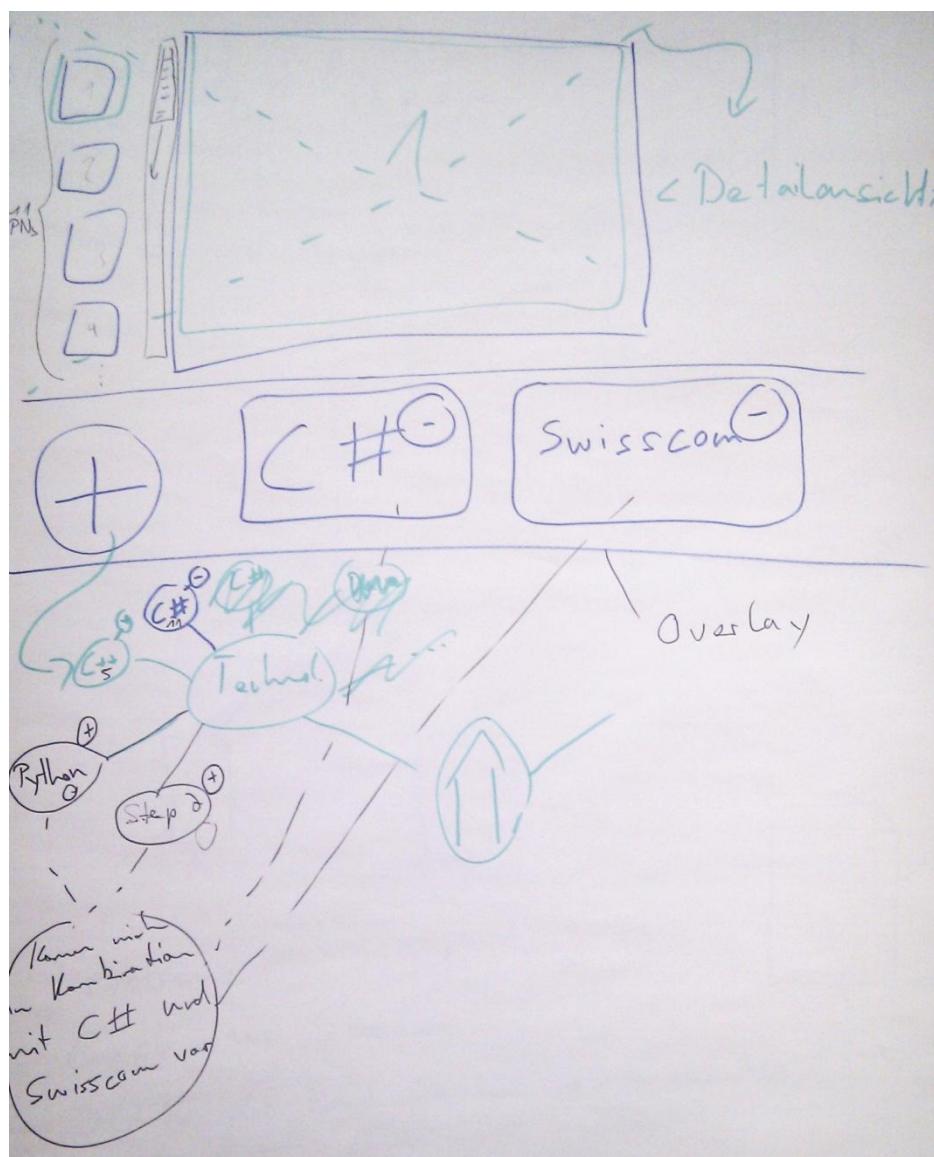


Abbildung 19 - Multi-Filter

Die Tree View zeigt alle übergeordneten Kategorien (Technologien, Firmen, Services, Tools etc.). Wird eine dieser Kategorien ausgewählt, so wird der Knoten mit Kindknoten erweitert, welche alle unter dieser Kategorie vorhandenen Begriffe anzeigen.

- Die Farbe Blau zeigt, welche Begriffe für die Filterung bereits ausgewählt sind. Ein erneutes Antippen eines solchen Knotens hebt die Auswahl auf (Wechsel zur Farbe Grün).
- Grün bedeutet, dass der Begriff zur Auswahl steht.

- Die schwarzen Knoten sind Merkmale, welche ebenfalls ausgewählt werden können. Werden sie ausgewählt, wird der Filter dann aber keine Ergebnisse liefern, da keine PN existiert, die allen diesen Angaben entspricht.

Die Begriffe, nach denen aktuell gefiltert wird, sind neben dem „Plus“ aufgelistet. Um einen Begriff aus dem Filter zu entfernen, muss das „Minus“ auf ebendiesem Begriff angetippt werden (ev. auch Drag & Drop möglich).

V.5.3.1.5 Finder

Anfangs werden im Fenster nur die Kategorien angezeigt. Wird eine ausgewählt, erscheint eine zweite Spalte, in der nun die zugehörigen Begriffe angezeigt werden. Die Auswahl wird somit hierarchisch aufgebaut und dem Nutzer wird immer klar angezeigt, wo er sich befindet.

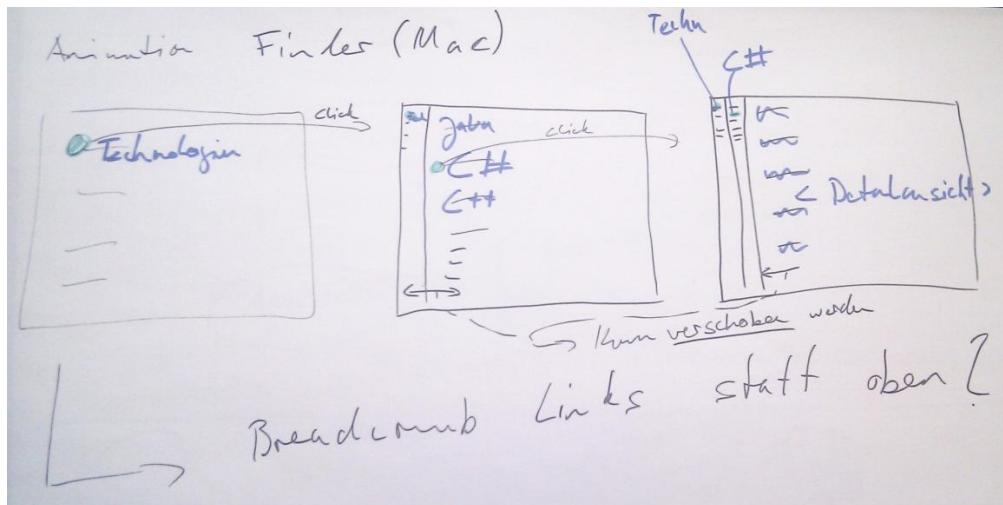


Abbildung 20 - Finder

V.5.3.1.6 Detailansichten

- List:** Normale Liste von Project Notes. Wird eine PN angetippt, so öffnet sich diese in einem Popup.
- Cover Flow:** In der Mitte wird das aktuelle Element angezeigt. Links und rechts sind die Vorgänger- bzw. Nachfolger-PN, durch welche geblättert werden kann.
- Scatter View:** Alle Project Notes liegen auf einem Haufen. Sie können einzeln verschoben und von Hand gruppiert werden. Ein Popup wird beim Anklicken einer PN geöffnet.
- Button Overview:** Die Project Notes werden als kleine Bilder aufgelistet. Es werden immer alle PN, die einer Kategorie zugeordnet sind, angezeigt. Daher kann für eine vergrösserte und somit detailliertere Ansicht mit Gesten ein Zoom ausgeführt werden.
- Browser:** Das aktuelle Element wird oben angezeigt, unten befindet sich die Liste aller Project Notes, in der navigiert werden kann (ähnlich wie Cover Flow).
- PowerPoint:** Die aktuelle Project Note wird gross angezeigt. Unterhalb der Grossansicht wird die Position der aktuell gewählten PN in der Liste aller PN angezeigt.



Abbildung 21 - Detailansichten

V.5.3.2 Papier-Prototyp

V.5.3.2.1 Variante A

In der ersten Variante wurde die Spider View umgesetzt (siehe Unterkapitel V.5.3.1.3 Spider View).

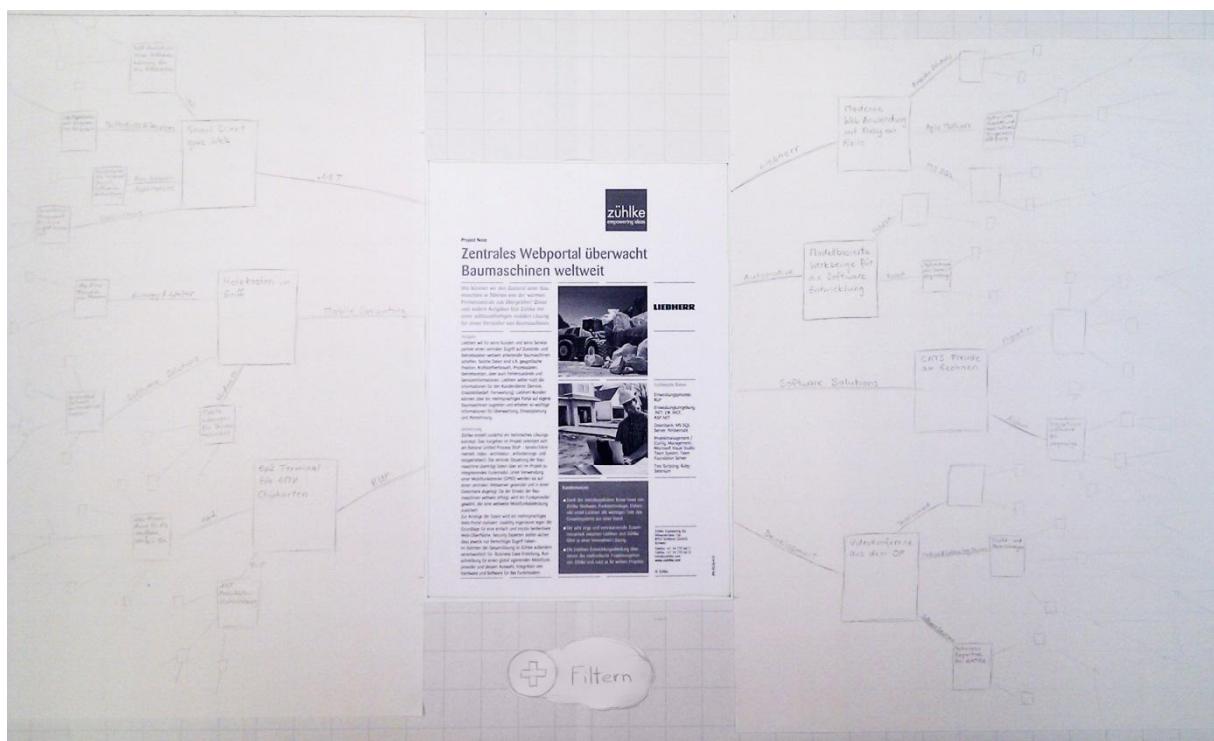


Abbildung 22 - Papierprototyp Variante A, Spider View

V.5.3.2.1.1 Testszenario

Das folgende Testszenario wurde mit Markus Stolze am 12.10.2011 getestet.

Sie sind als Mitarbeiter der Firma Swisscom spezialisiert auf Projekte im Bereich .Net. Heute haben Sie einen Termin bei der Zühlke Engineering AG, welche ihr Projektpartner ist. Es ist das erste Meeting, daher treffen Sie etwas verfrüh ein. Die Empfangsdame weist Sie darauf hin, dass die Project Notes der Firma (Projektbeschrieb auf eine A4-Seite zusammengefasst) auf dem neu erworbenen Surface eingesehen werden können.

Aufgabe 1

Sie nutzen daher die Wartezeit, um sich mit der neuen Technologie zu befassen. Gleichzeitig sind Sie neugierig herauszufinden, ob die Zühlke Engineering AG Erfahrungen in Ihrem Spezialgebiet mitbringt.

Aufgabe 2

Wie Sie feststellen können, hat es bisher einige Projekte mit .Net gegeben. Daher sind Sie nun daran interessiert, ob eines dieser Projekte in Zusammenarbeit mit Ihrem Arbeitgeber entstanden ist.

V.5.3.2.1.2 Fazit

Die Anzeige der Project Note ist zu klein. Um diese am Bildschirm angenehm lesen zu können, muss sie weiter vergrössert werden können.

Um festzustellen, welche Themen in der PN angesprochen werden, orientiert sich der Nutzer zuerst an den Informationen, die am rechten Rand auf der PN selbst aufgezeigt sind. Dabei kann es durchaus vorkommen, dass auf einer Project Note nicht alle Daten, welche zur Verfügung stehen, festgehalten sind. Somit kann es sein, dass Verbindungslienien mit Attributbeschreibungen, die auf der Project Note selbst nicht aufgelistet sind, für Verwirrung sorgen. Es wäre zudem logischer, wenn die Attribute nur rechts von der Project Note aufgelistet werden würden.

Der Filter unter der PN wird nicht wahrgenommen. Wollte die Testperson nun beispielsweise nach .Net Projekten suchen will, drückte sie auf die nächste Project Note, welche über die der Linie .Net verbunden ist. Zudem ging die Testperson davon aus, dass sie sich, nachdem sie der .Net Verbindungslienie gefolgt ist, in einem .Net Ast befindet und sich da nur PNs zu diesem Thema befinden. Dies ist aber nicht möglich, da jede PN nur einmal im Baum vorkommt. Die Navigation ist daher unverständlich für den Benutzer. Fall er spezifisch nach etwas suchen möchte, findet er seinen Weg nur über die Verbindungslienien allein nicht.

Dadurch, dass schon von Anfang an falsch navigiert wurde, konnten die Testszenarien nicht abgeschlossen werden. Es wurde ersichtlich, dass diese Variante verwirrend und irreführend für den Benutzer ist. Daher wurde sie als ungeeignet eingestuft.

V.5.3.2.2 Variante B

Die nächste Variante des Papier-Prototyps befasste sich mit der Umsetzung der Variante A-2 der Tree View. Dabei wurden aber einige Elemente anders dargestellt. Damit ersichtlich ist, wo man sich befindet, ist links neben der ausgewählten Project Note eine Box dargestellt, die mit der Oberkategorie beschriftet ist. Der ausgewählte Begriff wird als ein Stapel von Project Notes in der Box angezeigt. Rechts von dieser Box befindet sich die aktuell ausgewählte Project Note, welche auch noch weiter vergrössert werden kann. Darunter befindet sich ein Slider, mit welchem durch die weiteren Project Notes der aktuellen Kategorie navigiert werden kann. Am rechten Rand jeder Project Note werden die Kategorien angezeigt. Mit jeder Kategorie sind die jeweils zur Project Note gehörenden Begriffe durch eine Linie verbunden. Wird ein Begriff angetippt, so ändert sich die Beschriftung des PN-Stapels und gegebenenfalls auch die Beschriftung der Box selbst (oben links). Zudem werden die Project Notes in der Grossansicht und im Slider aktualisiert.

Wird jedoch am rechten Rand der Project Note auf eine Kategorie gedrückt, öffnet sich ein neues Fenster und die eben angetippte Kategorie bewegt sich in die Mitte des Fensters. Rundherum werden die dazu gehörigen Begriffe angezeigt und es kann einer davon ausgewählt werden. Dies ermöglicht dem Nutzer, dass er nicht nur Kategorien, die mit der aktuell angezeigten Project Note zusammenhängen, auswählen kann.

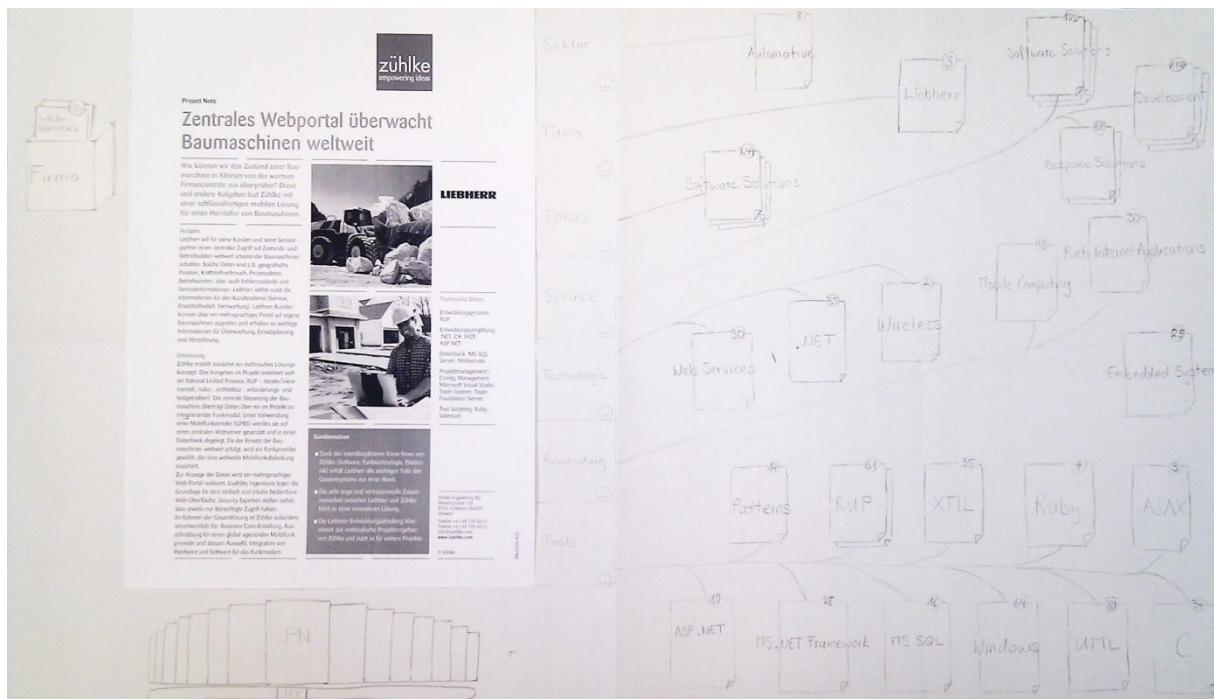


Abbildung 23 - Papierprototyp Variante B

V.5.3.2.2.1 Testszenario

Das folgende Testszenario wurde mit Kevin Gaunt am 14.10.2011 getestet.

Sie sind als Mitarbeiter der Firma Swisscom spezialisiert auf Projekte im Bereich .Net. Heute haben Sie einen Termin bei der Zühlke Engineering AG, welche ihr Projektpartner ist. Es ist das erste Meeting, daher treffen Sie etwas verfrüh ein. Die Empfangsdame weist Sie darauf hin, dass die Project Notes der Firma (Projektbeschrieb auf eine A4-Seite zusammengefasst) auf dem neu erworbenen Surface eingesehen werden können.

Aufgabe 1

Sie nutzen daher die Wartezeit, um sich mit der neuen Technologie zu befassen. Gleichzeitig sind Sie neugierig herauszufinden, ob die Zühlke Engineering AG Erfahrungen in Ihrem Spezialgebiet mitbringt.

Aufgabe 2

Einer Ihrer Arbeitskollegen erzählte Ihnen, dass er ein Projekt mit der Zühlke Engineering AG gemacht hatte. Sie mögen sich entfernt noch erinnern, um was es ungefähr ging, Ihnen fällt aber kein konkreter Begriff zu diesem Projekt ein. Trotzdem möchten Sie nun mehr über dieses Projekt herausfinden.

V.5.3.2.2.2 Fazit

Es ist irritierend, dass zu Beginn bereits eine Project Note ausgewählt ist. Der Nutzer weiß nicht, wie er dort hingekommen ist. Normalerweise wird zuerst eine Liste (Grid) mit einer Auswahl von PNs erwartet.

Der Slider unten links wird übersehen. Es ist besser ihn oberhalb der gross dargestellten Project Note zu platzieren.

Zusätzlich sollte für die Ansicht der PN ein Fokus Modus benutzt werden. Wenn man die Project Note genau lesen möchte, soll sie sich durch Antippen weiter vergrössern und alle anderen Informationen sollen in den Hintergrund rücken oder ausgeblendet werden.

Die Project Notes Stapel, welche auf der rechten Bildschirmhälfte angezeigt werden, sollten mit mehr als einer Note angezeigt werden. Sonst ist nicht verständlich, dass es sich bei der Zahl auf den gezeichneten Notes um die Anzahl Project Notes, die mit dieser Kategorie verbunden sind, handelt.

Die Testperson konnte die ihr gestellten Aufgaben lösen. Sie erwartete aber, dass nach dem Antippen einer Kategorie rechts neben der PN ein Menü mit nur den Begriffen, die zur Project Note gehören, angezeigt wird. Entgegen der Erwartung wurden alle unter der Kategorie vorhandenen Begriffe aufgelistet. Die Erwartung der Testperson entstand dadurch, dass die Kategorien mit der PN verbunden sind. Daher sollten die Kategorien am rechten Rand des Bildschirms aufgeführt werden und die Begriffe sollten mit der PN direkt verbunden. Die Trennung ist aber auch dann nicht ideal.

V.5.3.2.3 Variante C

Solange keine Benutzerinteraktion erfolgt, ist das Gerät im Demo Modus. Wird dieser beendet, erscheint eine Übersicht über alle PNs.

Wird davon eine Project Note angewählt, wird diese in einer Detailansicht (siehe Abbildung 24 - Papierprototyp Variante C, Detailansicht) angezeigt.

Mit dem Drücken der Pfeile lassen sich die nächsten bzw. vorhergehenden Project Notes anzeigen. Rechts werden immer die zur angezeigten Project Note gehörigen Begriffe aufgelistet.

Wird die grosse angezeigte Project Note angetippt, so findet ein Übergang in den Lesemodus statt. In diesem Modus wird die Project Note stark vergrössert und in der Mitte des Bildschirms positioniert angezeigt. Die mit der PN verbundenen Begriffe werden ausgeblendet, der Hintergrund wird abgedunkelt.

Wenn in der Detailansicht einer der mit der Project Note verbundenen Begriffe ausgewählt wird, so öffnet sich ein Kontextmenü, in welchem man den Begriff als Kriterium zum Filter hinzufügen kann. Falls schon ein oder mehrere Kriterien für den Filter gesetzt wurden, können diese entweder durch den Begriff als neues Filterkriterium ersetzt werden oder der Filter wird mit dem Begriff als weiteres Kriterium zusätzlich eingeschränkt.

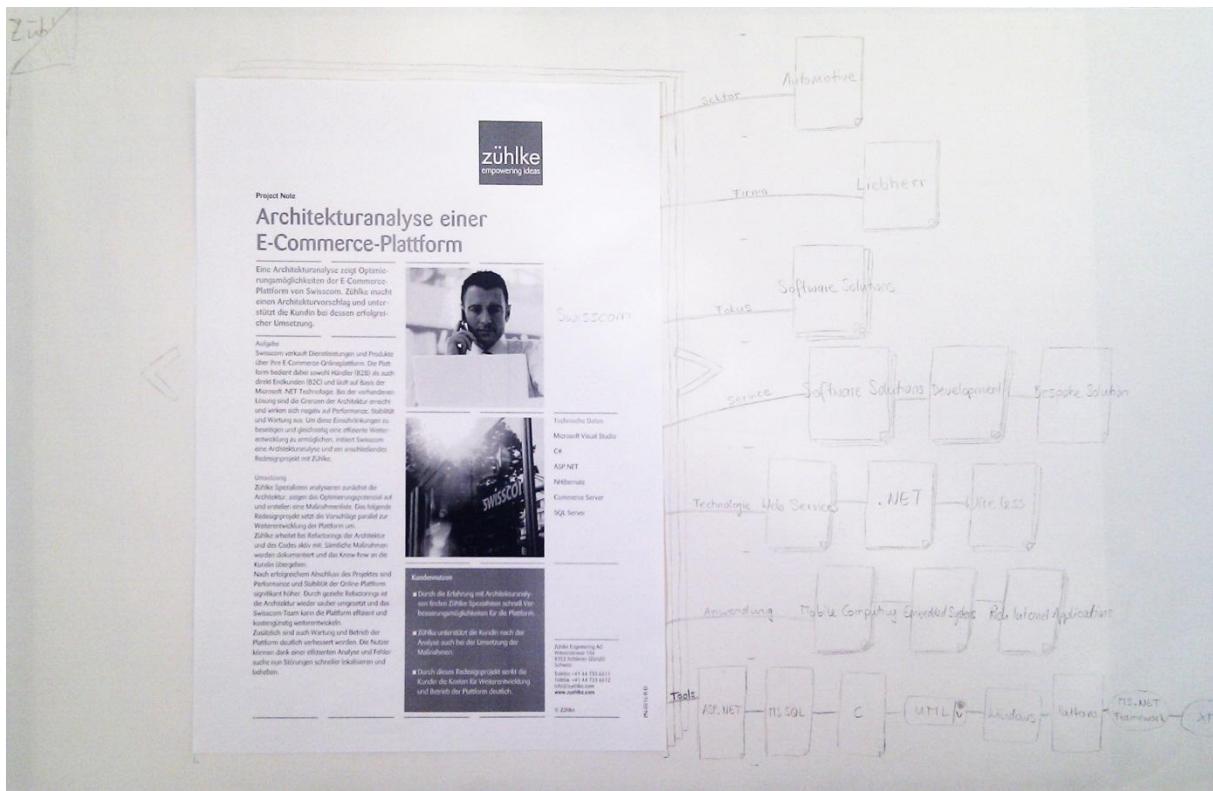


Abbildung 24 - Papierprototyp Variante C, Detailansicht

Nachdem der Filter weiter eingeschränkt oder neu gesetzt wurde, erfolgt ein Wechsel zur Gesamtübersicht (siehe Abbildung 25 - Papierprototyp Variante C, Übersicht). Um eine klare Übersicht über alle Project Notes zu erhalten, wird hier eine Liste von PNs angezeigt. Oben links wird angegeben, wie viele PNs momentan sichtbar sind. Gleich daneben können Filterkriterien eingegeben werden. Mit einem Klick auf das „Plus“ öffnet sich eine weitere Liste (im Stil eines Finders, siehe Unterkapitel V.5.3.1.5 Finder, oder einem Drop Down Menu), aus der ein Begriff ausgewählt werden kann. Sobald ein Filter gesetzt ist, passt sich die Liste der Project Notes an.

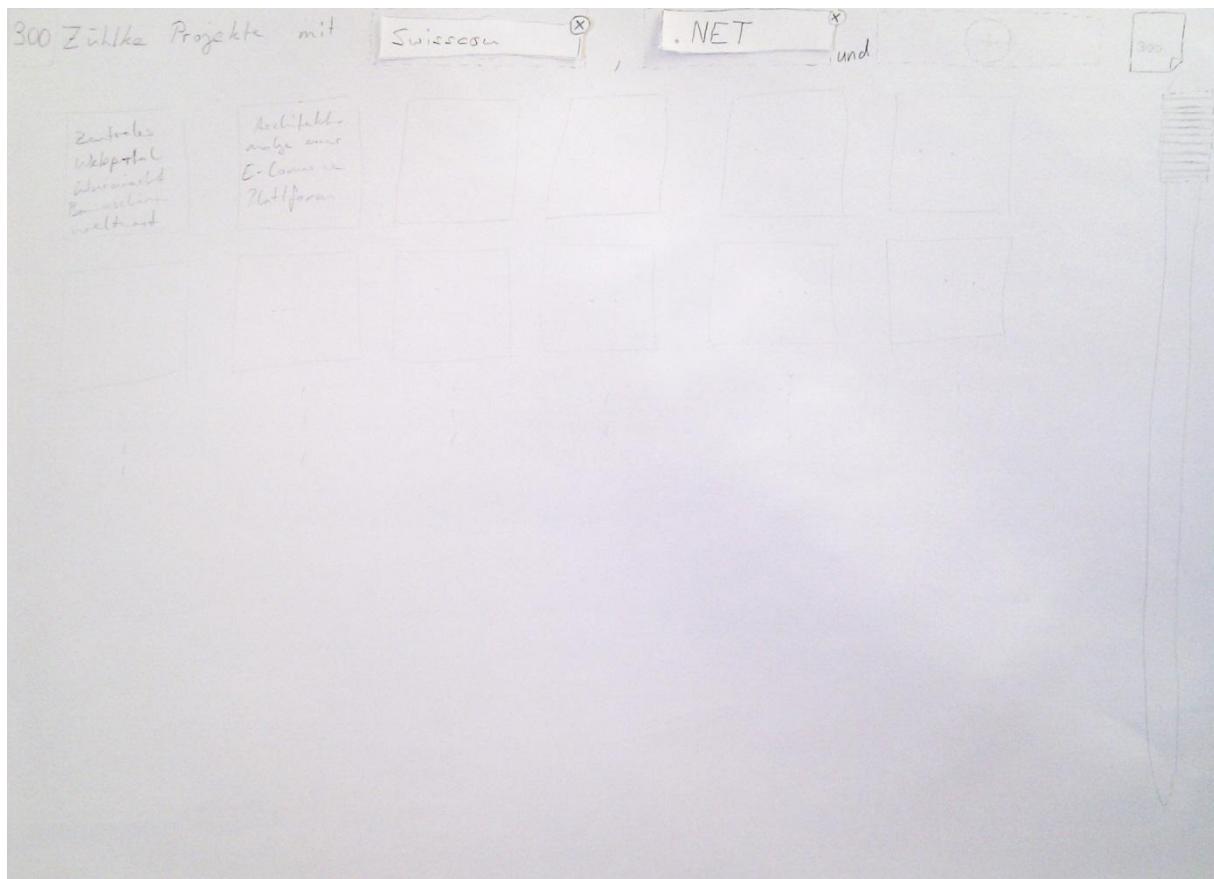


Abbildung 25 - Papierprototyp Variante C, Übersicht

Wählt man eine Project Note aus der Liste aus, öffnet sie sich in der Detailansicht. Möchte man dann von der Detailansicht wieder zur Gesamtübersicht zurückkehren, so kann dazu oben links die umgeknickte Ecke nach unten gezogen werden.

V.5.3.2.3.1 Testszenario

Das folgende Testszenario wurde mit Silvan Gehrig am 14.10.2011 und Christian Moser und Marco Balzarini am 17.10.2011 getestet.

Für das Testen dieses Papier-Prototyps wurde ein bereits bestehendes Testszenario verwendet (siehe Unterkapitel V.5.3.2.1.1 Testszenario) und um die nachfolgende Aufgabe erweitert.

Aufgabe 3

Sie möchten sich nun nochmals einen Überblick über alle Projekte verschaffen.

V.5.3.2.3.2 Fazit

Der Demo Modus muss genauer ersichtlich sein, er soll bereits erkannt werden, wenn man sich an den Surface setzt. Ansonsten beginnt man zu lesen und die Project Note wird plötzlich gewechselt. Der Demo Modus muss zudem allgemein überdenkt werden. Es wäre auch ein klassischer Bildschirmschoner denkbar.

Der angezeigte Slider bei der Übersicht ist nicht nötig.

In der Detailansicht wäre es gut, wenn die Begriffe so dargestellt sind, dass es offensichtlich ist, dass sich bei dessen Betätigung ein Kontextmenü öffnet. Zudem müssen die Filterkriterien auch in der Detailansicht ersichtlich sein (am oberen Rand).

Die Testperson fand schnell mehrere Möglichkeiten, um den Filter hinzufügen zu können. Nach kurzem Auseinandersetzen mit dem Fenster, welches eine Überblick über die Project Notes bietet, fand sie sich gut zurecht. Sie konnte alle gestellten Aufgaben in angemessener Zeit lösen.

Ergänzungen aus dem Test vom 17.10.2011 mit Christian Moser und Marco Balzarini:

Die umgeknickte Ecke soll weggelassen werden. Besser ist es, die Auswahl an PN's immer noch abgedunkelt im Hintergrund zu sehen. Die Detailansicht „legt“ sich dann über die Gesamtübersicht.

Da zwei Typen von Personas bestehen, könnte für den Bewerber einen anderen Einstieg ins Programm gewählt werden. Die Unterscheidung zwischen den Personas könnte mit Badges, welche je nach Besuchertyp anders getagged sind, realisiert werden. Um den Surface nutzen zu können, muss der Badge auf den Bildschirm gelegt werden. Die getaggten Objekte rufen dann verschiedene Informationen auf, welche sich auf die jeweilige Persona fokussieren. Diese Idee wird in der Umsetzung vorerst ignoriert.

Neben kleinen Verbesserungen ist diese Variante bestens geeignet für das Projekt und wird daher umgesetzt.

V.5.3.3 Design Entscheide

V.5.3.3.1 Pivot Viewer

Der Pivot Viewer von Steve Marx [marx10] würde eine Möglichkeit bieten, die Project Notes auf dem Surface darzustellen und einen einfachen aber sogleich attraktiven Filter anzubieten.

Pivot View of Netflix Instant Watch Movies

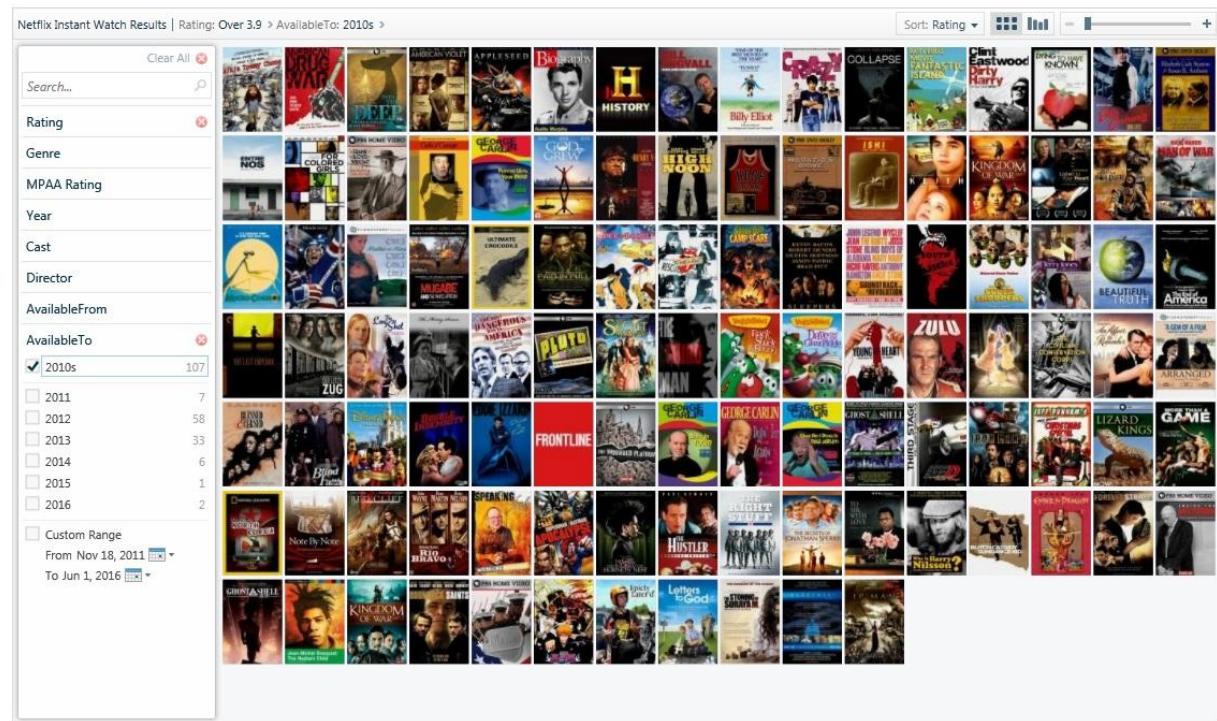


Abbildung 26 - Pivot Viewer

Nun sprechen aber einige Punkte gegen die Verwendung des Pivot Viewers:

- Nicht passend für Project Flip 2.0 ist die Tatsache, dass die einzelnen Elemente bei grosser Anzahl stark verkleinert dargestellt werden, damit alle gleichzeitig auf dem Bildschirm sichtbar sind. Dadurch wäre der Titel der Project Note nur noch schlecht oder gar nicht mehr erkennbar. Die Lesbarkeit des Titels sowie auch eine angenehme Grösse des dargestellten Bildes sind jedoch wichtig für die Entscheidungsfindung des Benutzers.
- Die ständige Anzeige aller zur Auswahl stehender Filterkriterien nicht erwünscht ist.
- Ferner ist die Detailansicht aus Performancegründen nicht gut geeignet. In dieser Ansicht werden auch die Elemente, welche sich gleich neben dem aktuellen Element befinden, angezeigt. Für Project Flip 2.0 würde das bedeuten, dass in der Detailansicht bis zu neun Project Notes gleichzeitig geladen werden müssten. Dies würde eine Weile dauern und so deutlich den Programmfluss stören.

V.5.3.3.2 Papierprototyp

Folgende Änderungen wurden gegenüber dem Papierprototyp, welcher im Unterkapitel V.5.3.2 Papier-Prototyp beschrieben wurde, durchgeführt:

Der Microsoft Surface 2 Touchtisch verfügt über eine Grosse Breite im Vergleich zur Höhe. Daher wurde der Filter auf der linken Seite platziert. Zur Entfernung eines Filters steht keine spezielle Schaltfläche mehr zur Verfügung, dies wird nun durch einmaliges Klicken auf ein Filterkriterium durchgeführt.

Die Anzahl der Projekte wird über den Filterkriterien aufgelistet.

Die Projektmetadaten werden als Projektdaten aufgelistet und befinden sich links neben der Project Note, damit sie sich in der Nähe des Filters befinden.

Die Scrollbar ist nur sichtbar, wenn gescrollt wird.

V.5.3.4 Externes Design

Nach den in Unterkapitel V.5.3.3 Design Entscheide aufgelisteten Design Entscheiden, wurde das Externe Design erarbeitet. Bei dem externen Design wurden das Schriftbild und die Farbpalette der Zühlke Engineering AG mehrheitlich übernommen. Die verkleinerten Project Notes orientieren sich optisch an der Grossansicht der PN.

Am linken Rand befindet sich der Filter, welcher in jeder Ansicht sichtbar ist. Die drei leeren Slots geben an, wie viele Filterkriterien maximal gesetzt werden können. Über den Knopf „Kriterien hinzufügen“ können die gewünschten Filterkriterien ausgewählt und hinzugefügt werden. Das Hinzufügen geschieht entweder über das Antippen des Kriteriums oder das Herüberziehen desselben in einen Slot. Will man einen Begriff entfernen, kann er erneut angetippt oder aus dem Slot gezogen werden.

Sind die Slots schon belegt und es wird trotzdem ein weiteres Kriterium aus der Liste ausgewählt, so bewegt sich dieses ein kurzes Stück in Richtung der Slots und dann aber wieder zurück an seinen Platz in der Liste. Damit wird der Nutzer darauf aufmerksam gemacht, dass zuerst ein Kriterium entfernt werden muss, bevor ein weiteres gesetzt werden kann.

Oberhalb der Filterkriterien wird immer angezeigt, wie viele Project Notes gerade aufgelistet werden und wie viele gesamthaft verfügbar sind.

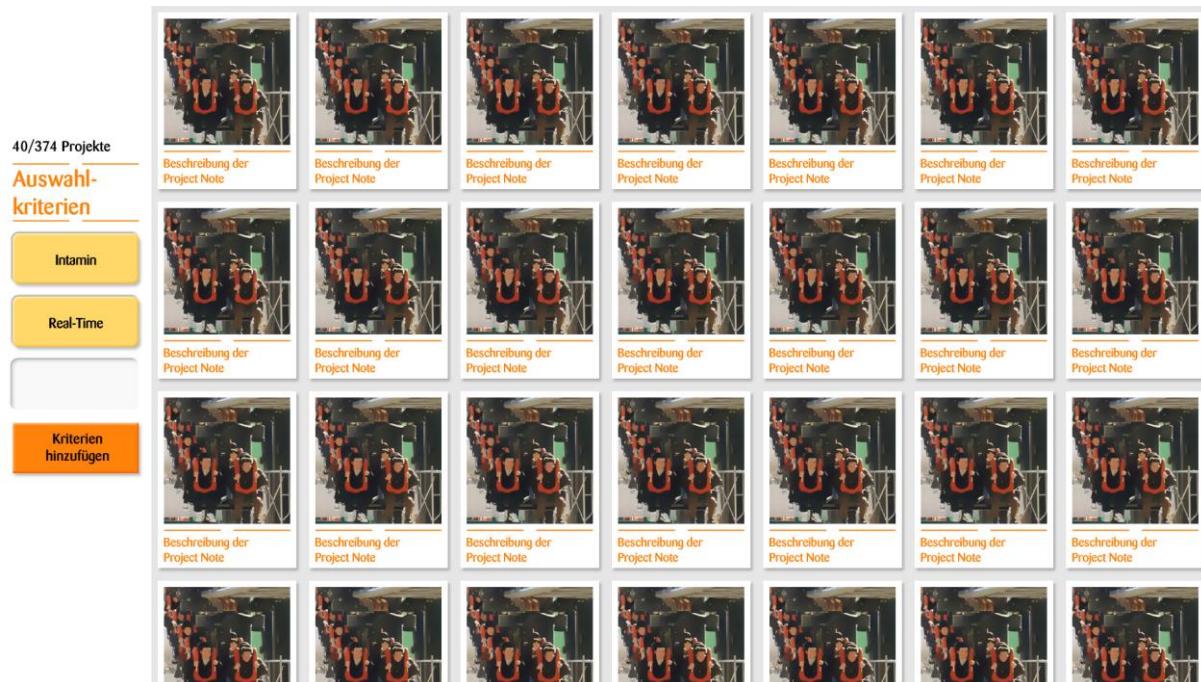


Abbildung 27 - Externes Design, Übersicht

Wird auf eine verkleinerte PN gedrückt, so öffnet sich diese in der Detailansicht (siehe dazu Abbildung 28 - Externes Design, Detailansicht). Am linken Rand der Project Note werden die ihr zugehörigen Projektdaten angezeigt. Sofern man sich für einen dieser Begriffe als Filterkriterium interessiert, können auch diese ausgewählt und dem Filter hinzugefügt werden.

Die Pfeile ermöglichen die Navigation zur nachfolgenden oder vorhergehenden Project Note in der Liste.

Um die Project Note gut lesen zu können, wird die Möglichkeit geboten, diese mithilfe einer Zoom Geste zu vergrößern.

Das Schliessen der Detailansicht geschieht durch das Berühren der abgedunkelten Fläche.

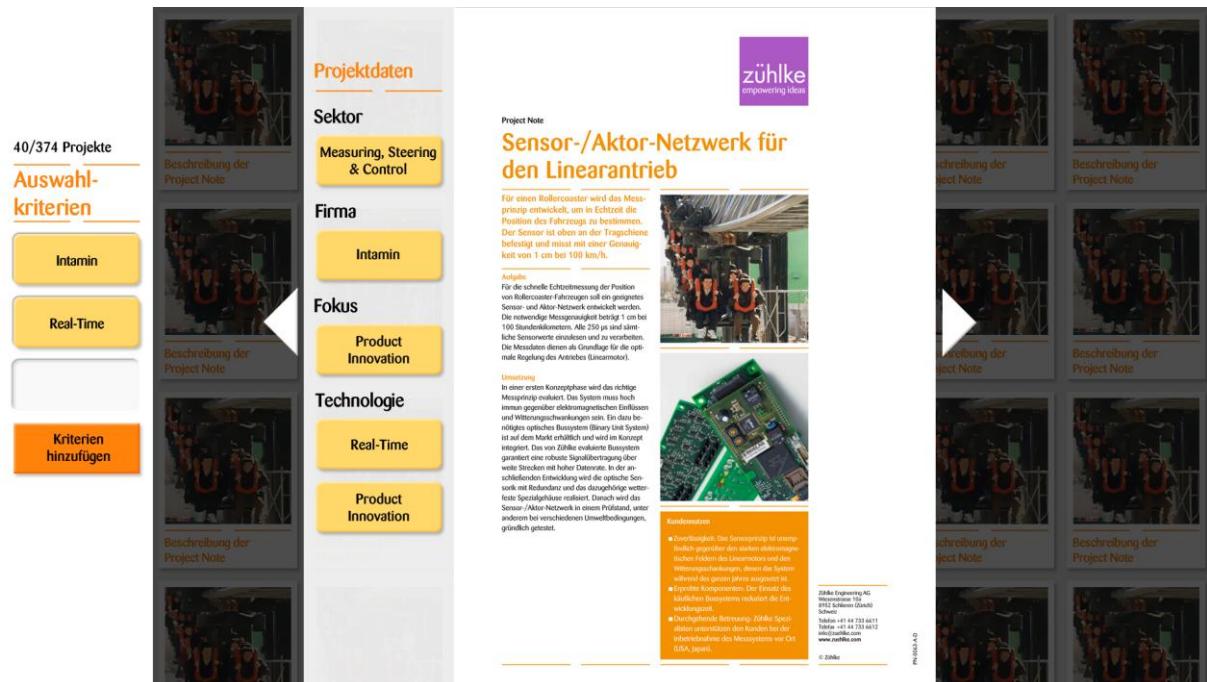


Abbildung 28 - Externes Design, Detailansicht

V.5.3.5 Guidelines

Der „Microsoft Surface 2.0 Design and Interaction Guide“³ definiert eine Reihe von Prinzipien, die wenn möglich eingehalten werden sollten. Project Flip 2.0 besitzt eine Reihe von Anforderungen, die sich nicht mit den Guidelines vereinen lassen. Die nicht eingehaltenen Richtlinien werden hier aufgeführt.

V.5.3.5.1 Interaction Design Guidelines

Section 3.1 Punkt 1:

„1. Create experiences for several people to use at the same time

*Microsoft Surface recognizes and responds to over 50 different touches at the same time. It sees fingers and objects touching the screen. This enables several people to gather around Surface and share applications, elevating solitary activities to social experiences.*³

Die Applikation ist primär nur auf einen Nutzer ausgerichtet. Die Anzeige und das spätere Lesen einer Project Note benötigt viel Platz auf dem Bildschirm, weshalb nur jeweils eine PN in der Detailansicht und dem Lesemodus angezeigt wird. Daher wird es immer nur für eine Person möglich sein, die Applikation zu benutzen. Es kann zwar vorkommen, dass sich zwei Nutzer zusammen eine PN anschauen. Jedoch nicht, dass die beiden unterschiedliche Aktionen durchführen. Dadurch fallen die unter Section 3.1 Punkt 1 aufgelisteten Kriterien weg.

³ [microsoft11] Microsoft Corporation, „Microsoft Surface 2.0 Design and Interaction Guide“, Juli 2011, <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=26713>, letzter Zugriff: 04.11.2011

Besonderes Augenmerk gilt zudem der Guideline

Section 3.1 Punkt 1.a:

„a. Use 360° degree application design for horizontal deployments

Surface recognizes touch orientation – it sees which direction fingers and special objects are pointed as they contact the screen. This enables developers to generally determine which side of the screen a particular person is on. People will use Surface from all sides so it's important that horizontal Surface deployments be designed for 360° usage.“³

Da immer nur ein Nutzer die Applikation bedient und sich der Surface vermutlich vor einer Sitzgruppe befindet, wird die Applikation sich nur auf eine Richtung ausrichten. Optional ist eine Drehung um 180° vorgesehen.

Section 3.1 Punkt 2.a:

„a. Direct touch interactions and indirect touch interactions

Direct touch interactions are physical movements of virtual content within the application by a finger or physical object. Indirect touch interactions usually rely on application interface chrome or abstract gestures. Examples of indirect touch interactions can include buttons, sliders, menus, and gesturing with symbol drawing. Direct touch interactions are the preferred type of interaction for use in Surface because they help to create more intuitive, content oriented experiences.“³

Das Hinzufügen der Kriterien zum Filter könnten auch per Drag & Drop realisiert werden. Da aber prinzipiell immer zwei verschiedene Filtermöglichkeiten bestehen (1) nur nach neuem Kriterium filtern oder 2) Kriterium dem Filter zusätzlich hinzufügen), ist die Variante eines Buttons mit ausklappbaren Menu einfacher bedienbar.

Section 3.1 Punkt 3:

„3. Use physical objects to enhance the experience“³

Die Hardware ist erst ab Januar 2012 verfügbar, daher können Interaktionen mit Objekten nicht getestet werden und werden daher weggelassen.

Section 3.1 Punkt 5.a:

„a. Make content the interface

- Do not replace direct touch interactions with UI controls such as buttons, menus and sliders.“³*

Siehe Section 3.1 Punkt 2.a

Section 3.1 Punkt 6.a:

„a. Transitions must be fluid and smooth

Smooth transitions give the user context about where they are in the experience.“³

Durch den zeitlich begrenzten Rahmen haben die Animationen eine tiefe Priorität erhalten. Sofern keine Verzögerungen bei der Implementation entstehen, sollten diese jedoch umgesetzt werden können.

V.5.3.5.2 Visual & Motion Design Guidelines

Section 3.2 Punkt 1.b:

„b. Grid-free layouts

Most visual designers have learned to create layouts based on grids. The 360° degree nature of Surface is great for laying out applications without a global, or screen wide, grid system. This requires a fresh perspective on visual layout; the Surface SDK ScatterView control is a quick and easy way to create grid-free global layouts. It encourages people to organize and explore content. ScatterView acts as an invisible container for onscreen objects, enabling some content to be oriented towards each edge of the screen by default, which encourages curiosity, direct touch interactions, and exploration.“³

Für eine übersichtliche Anzeige der Project Notes ist ein Raster von Nöten. Platzierte man diese in der erwähnten ScatterView, würde dies zu einer chaotischen Anzeige führen.

Section 3.2 Punkt 2.b:

„b. Creating depth using 3D

True 3D uses rendered three-dimensional geometries in real time. This enables realistic rotation of cubes, spheres, custom 3D models, and so on. While 3D can be authored in XAML and delivered using WPF, it can adversely impact application performance. True 3D is best created and delivered on Surface using XNA, the core Microsoft 3D and gaming engine.”³

3D Elemente könnten in einer Erweiterung verwendet werden, haben aber für dieses Projekt keinerlei Priorität und werden daher nicht umgesetzt.

Section 3.2 Punkt 8:

„8. Motion Design

Motion design defines how things move onscreen, a critical part of the Surface experience. Animations always support the content and the experience as a whole. Motion design provides animations and effects that conveys emotion, energy, connection, and responsiveness. They provide visual hints, cues, and an invitation to explore content.”³

Siehe Section 3.1 Punkt 6.a

Project Flip 2.0

Studienarbeit

V.6 Entwurf

V.6.1 Dokumentinformationen

Datum	Version	Änderung	Autor
25.09.2011	1.0	Erste Version des Dokuments	lelmer
03.10.2011	1.1	Review	dreichl
12.12.2011	1.2	Schichtenarchitektur dokumentiert	lelmer
15.12.2011	1.3	Architektur angepasst, Sensitivitätsanalyse	lelmer
16.12.2011	1.4	Review	cheidt
19.12.2011	1.5	Review	dreichl
19.12.2011	1.6	Systemübersicht aktualisiert	lelmer
20.12.2011	1.7	Weitere Design Entscheide	lelmer
20.12.2011	1.8	Review	dreichl
21.12.2011	1.9	Korrekturen	cheidt

V.6.2 Design Entscheide

V.6.2.1 PDF auf Surface 2 Darstellen

Die primäre Aufgabe der Software ist es, PDF Dokumente auf dem Surface 2 darzustellen. Dazu gibt es verschiedene Möglichkeiten, die genauer analysiert wurden. Zur Entscheidung wurde schliesslich eine Nutzwertanalyse durchgeführt.

V.6.2.1.1 Varianten

V.6.2.1.1.1 Variante 1: PDF direkt darstellen

Eine naheliegende Lösung ist natürlich, das PDF direkt auf dem Surface darzustellen. Leider unterstützt das Surface 2 Framework diese Darstellung nicht direkt; es müsste noch eine PDF Library zur Darstellung zur Laufzeit geladen werden. Dadurch wird die Laufzeitumgebung von einer zusätzlichen Library abhängig, was die Installation erschwert oder gar verunmöglicht. Dies kann zurzeit nicht genauer beurteilt werden, da die zu testende Hardware noch nicht verfügbar ist.

V.6.2.1.1.2 Variante 2: Umwandlung zu XPS

Das Surface 2 SDK erlaubt es, ohne weitere Libraries XPS Dokumente anzuzeigen. Dies geschieht mit Hilfe der DocumentViewer Klasse. Das XPS Dokument ähnelt im Aufbau einem PDF Dokument, so können Schriften und Vektorgrafiken beliebig vergrössert werden, ohne dass dabei ein Qualitätsverlust entsteht.

V.6.2.1.1.3 Variante 3: Umwandlung zu Bild

Die einfachste Möglichkeit, um ein PDF auf dem Surface 2 darzustellen, ist eine Umwandlung des PDF Dokuments in eine Rastergrafik. Diese können ohne Probleme auf dem Surface 2 dargestellt werden und auch die Umwandlung eines PDFs zu einem Bild ist mit Open Source Libraries problemlos möglich (z.B. mit Image Magick oder GhostScript/GhostPDF/GhostXPS). Der grosse Nachteil besteht darin, dass die Bilder sehr viel Speicherplatz verbrauchen würden. Eine Rastergrafik verfügt zudem über eine Maximalgrösse, wird diese bei einer Vergrösserung überschritten, kann das Bild nicht mehr verlustfrei angezeigt werden.

V.6.2.1.2 Nutzwertanalyse

In der Nutzwertanalyse (siehe Tabelle 5 - Nutzwertanalyse) geht die **Variante 2: Umwandlung zu XPS** als Sieger vor der Variante 3: Umwandlung zu Bild hervor.

Nutzwertanalyse: PDF auf Surface 2 darstellen							
Kriterium	Gewichtung	Variante 1		Variante 2		Variante 3	
		PDF direkt darstellen	Umwandlung zu XPS	Bewertung	Total	Bewertung	Total
		1 bis 5	1 bis 5	1 bis 5		1 bis 5	
Programmieraufwand	2	1	2	3	6	5	10
Kopplung Laufzeitumgebung	5	1	5	5	25	5	25
Testbarkeit	4	1	4	5	20	5	20
Darstellungsqualität	4	5	20	4	16	1	4
Total Punkte			31		67		59
Rang			3		1		2

Bemerkung: Höhere Gewichtungen / Bewertungen sind besser.

Tabelle 5 - Nutzwertanalyse

V.6.2.1.3 Sensitivitätsanalyse

In der Sensitivitätsanalyse wird untersucht, wie stark sich eine Änderung auf das Gesamtergebnis auswirken würde.

Die Variante 1 wird auch bei Änderungen der Bewertung nicht als Sieger hervorgehen.

Zwischen der Variante 2 und 3 ist die Darstellungsqualität das entscheidende Kriterium. Eine andere Bewertung dieses Kriteriums würde das Ergebnis so stark beeinflussen, dass die Variante 3 zum Gewinner würde. Dieses Kriterium ist jedoch für die Zühlke Engineering AG sehr wichtig. Zudem wurde die Variante 3 bereits im Vorgängerprojekt Project Flip 1.0 verfolgt, leider mit einem nicht vollständig zufriedenstellenden Ergebnis. Somit ist die Wahl der **Variante 2: Umwandlung zu XPS** klar begründet.

V.6.2.2 Textdatei für Aggregation statt Datenbank

Gemäss Anforderungen muss ein Mapping der Metadaten möglich sein, da die Rohdaten zu stark gestreut sind. So muss z.B. der Begriff „JBoss“ auf Java gemappt werden können. Es wurde entschieden, dass dieses Mapping in einer Textdatei gespeichert wird, um so auf eine Datenbank verzichten zu können. Denn dadurch wird die Systemkomplexität verringert. Zusätzlich ist es leicht möglich, das Mapping mit einem Tabellenkalkulationsprogramm wie Excel zu bearbeiten.

V.6.2.3 Verwendung des Downloaders als Interface

Da die Applikation nicht in der Zühlke Engineering AG entwickelt wurde, konnte nicht auf den internen Sharepoint Server oder auf die interne Dateiablage zugegriffen werden. Da die PDF Dateien trotzdem heruntergeladen werden mussten, wurde dazu der öffentliche Webserver der Firma verwendet. Um die Software in der Zühlke Engineering AG betreiben zu können, muss noch ein Downloader entwickelt werden, der die PDF Dateien vom internen Server herunterlädt. Um diese zukünftige Implementation zu vereinfachen, wurde für den im Projekt verwendeten Downloader an so vielen Orten wie möglich ein Interface statt der konkreten Implementation verwendet, damit der aktuelle Downloader leicht mit dem neuen, zukünftigen Downloader ausgetauscht werden kann.

V.6.3 Architektur

Bei der Architektur wurde darauf geachtet, dass die einzelnen Komponenten möglichst unabhängig sind und so einfach ausgetauscht werden können.

V.6.3.1 Systemübersicht

Das System ist in mehrere Komponenten unterteilt. Die drei Hauptkomponenten sind:

- Sharepoint Server
- Webserver der Zühlke Engineering AG
- Microsoft Surface 2

Hier ein Überblick über die Komponenten und deren Rollen:

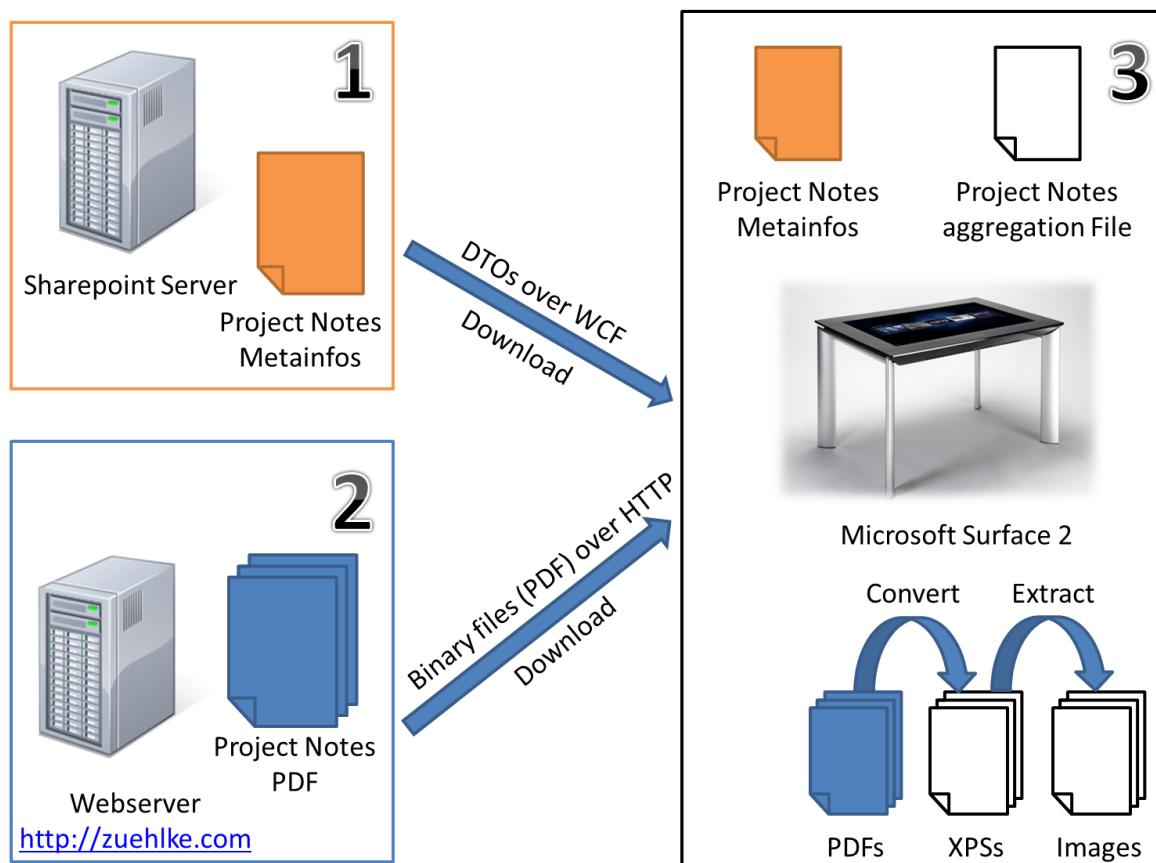


Abbildung 29 - Systemübersicht

Quelle Microsoft Surface 2 Bild: <http://blogs.msdn.com/b/surface/archive/2011/11/17/preorder.aspx>, letzter Zugriff: 20.12.2011

V.6.3.1.1 Sharepoint Server (1)

Auf dem internen Sharepoint Server der Zühlke Engineering AG werden die Project Notes und deren Metainformationen verwaltet. Nachdem eine neue Project Note publiziert wurde, werden die Metadaten erfasst und auf dem Sharepoint Server abgelegt. So können verschiedene Personen und Applikationen diese Daten abrufen und verarbeiten.

Bezüglich Metadaten ist hervorzuheben, dass diese nicht direkt verwendet werden können, da sie zu uneinheitlich und zu stark gestreut sind. Deshalb müssen diese Metadaten noch aggregiert werden, was in einem nächsten Schritt auf dem Surface 2 geschieht (siehe Unterkapitel V.6.3.1.3 Microsoft Surface 2 (3)).

In der aktuellen Implementation besteht keine Anbindung an den Sharepoint Server, da die Hardware zum Zeitpunkt des Projektes noch nicht zur Verfügung stand und von Extern kein Zugriff bestand.

V.6.3.1.2 Webserver der Zühlke Engineering AG (2)

Der Webserver hostet die Internetseite der Zühlke Engineering AG. Zudem stehen dort auch diverse Project Notes zum Download zur Verfügung.

Zu Beginn der Arbeit wurde von der Zühlke Engineering AG ein statisches Textdokument bereitgestellt, in welchem die Metainformationen aller Project Notes vorhanden sind. Mit Hilfe dieser Informationen können die PDF Dokumente direkt vom Webserver heruntergeladen und gespeichert werden. So stehen sie für die Applikation zur Verfügung.

V.6.3.1.3 Microsoft Surface 2 (3)

Der Microsoft Surface 2 wird in der Eingangshalle der Zühlke Engineering AG stehen, damit wartende Kunden sich mit den Project Notes auseinander setzen können. Um die Verfügbarkeit des Services sicherzustellen, muss der Surface möglichst unabhängig von den anderen Komponenten sein. Deshalb werden die für die Applikation benötigten Informationen periodisch heruntergeladen und verarbeitet. Dieser Vorgang ist von der Programm-Ausführung getrennt und wird im Betriebskonzept (siehe V.8.3.1 Betriebskonzept) genauer beschrieben.

Bevor die Applikation gestartet werden kann, müssen die gespeicherten PDF Dokumente weiter verarbeitet werden. Als erstes werden die PDF Dokumente in XPS Dokumente konvertiert. In einem zweiten Schritt werden Bilder aus der XPS Datei extrahiert.

Die Metadaten der Project Notes enthalten Bezeichnungen, welche sehr vielfältig sind (z.B. Java, JBoss, JavaBeans, JPA). Eine wichtige Anforderung an das Projekt war die Benutzerfreundlichkeit der Applikation, weshalb sichergestellt werden musste, dass nicht zu viele Informationen auf einmal dargestellt werden.

Aus diesem Grund wurde ein Mapping entwickelt, wodurch verschiedene Namen auf einen gemeinsamen Begriff (z.B. Java) zu mappen. Technisch wurde dies so realisiert, dass vor dem Einlesen der Metadaten zuerst eine andere Datei für das Mapping eingelesen wird. Das Mapping wird dann in einem Dictionary gespeichert, das könnte beispielsweise folgendermassen aussehen:

Java	Java
JBoss	Java
JavaBeans	Java
JPA	Java
ADO.NET	.NET
WPF	.NET

Tabelle 6 - Beispiel Mapping Dictionary

Dieses Mapping wird in einer mit Tabulatoren getrennten Textdatei gespeichert, damit diese einfach mit einem Tabellenkalkulationsprogramm wie Excel bearbeitet werden kann. Um die Bearbeitung zusätzlich zu vereinfachen, können mehrere Begriffe auf eine Zeile geschrieben werden, die dann alle auf das erste Keyword in der Zeile gemappt werden. Ein Beispiel dafür:

Java	Java	JBoss	JavaBeans	JPA
.Net	ADO.Net	WPF		

Tabelle 7 - Beispiel Mapping Datei

Nachdem das Mapping eingelesen wurde, werden die Metadaten eingelesen. Da das Mapping in einem Dictionary gespeichert ist, können die einzelnen Bezeichnungen performant auf die ausgewählten Begriffe gemappt werden.

Der Vorteil dieser Lösung ist, dass die Metadaten nicht von Grund auf aufgearbeitet oder verändert werden müssen, sondern diese Änderungen ausschliesslich in der Mapping Datei vorgenommen werden müssen. So kann die neue Metadaten Datei jeweils einfach vom Sharepoint Server heruntergeladen werden und die alte Datei überschreiben.

V.6.3.2 Klassendiagramm mit Fokus Project Note / Metadata

Das Wichtigste an Project Flip 2.0 sind die Projektreferenzen, also die Project Notes. Deshalb wird hier dargestellt, wie die Project Notes mit den Metadaten zusammenhängen:

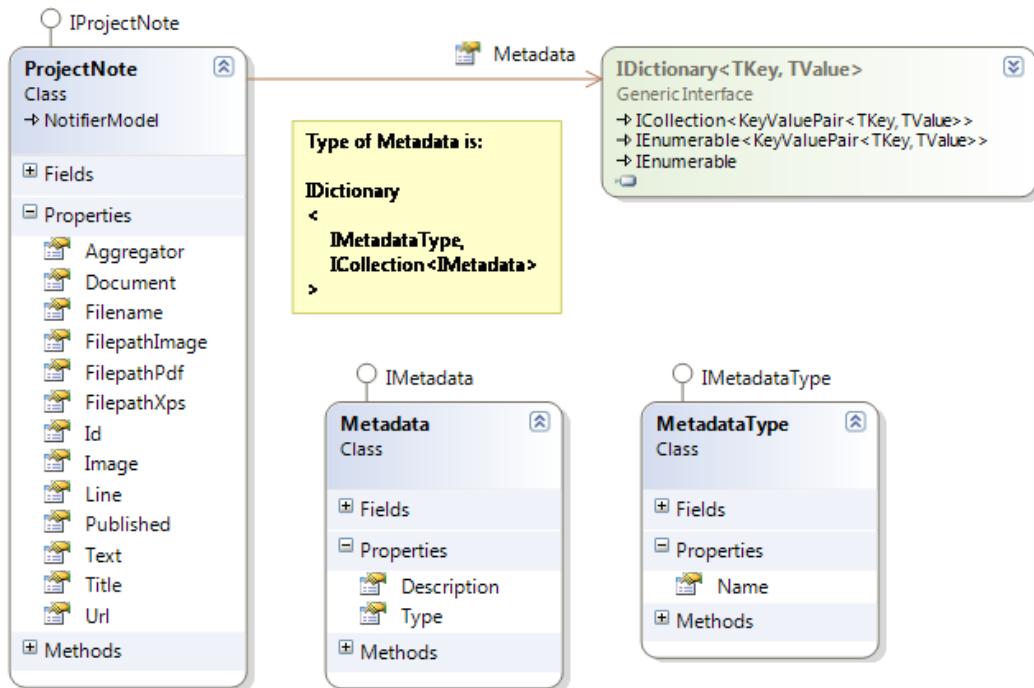


Abbildung 30 - Klassendiagramm mit Fokus Project Note / Metadata

Eine ProjectNote hat also ein Dictionary Metadata, das als Key einen MetadataType enthält. Mit diesem Key kann dann auf eine Liste von Metadaten zugegriffen werden. Der Vorteil dieser Lösung ist, dass nicht nur die Metadaten sondern auch die Metadaten Typen sich beliebig ändern können und z.B. auch aggregiert werden können. So kann man auch auf eine einfache Art eine neue Art zur Filterung hinzufügen, indem ein neuer Metadaten Typ hinzugefügt wird.

V.6.3.3 Assemblies und Namespaces

Das Projekt enthält, im Vergleich zu anderen, wenige Zeilen Code. Der Code könnte daher schon als overengineered bezeichnet werden. Dies wurde aber explizit von der Zühlke Engineering AG so gewünscht, da das Projekt noch weiterentwickelt werden soll.

Zum Beweis der Schichtenarchitektur wurde im Visual Studio 2010 ein Dependency Diagramm erstellt. Wie dabei vorgegangen wurde, ist in Abbildung 31 - Generierung Dependency Graph ersichtlich.

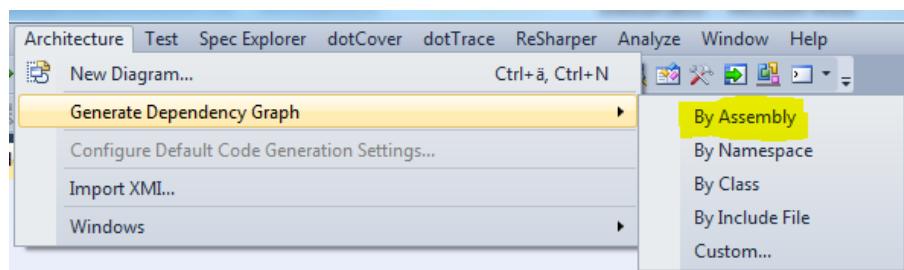


Abbildung 31 - Generierung Dependency Graph

Um das Diagramm übersichtlich zu halten, wurden in einem zweiten Schritt die „Generics“ und die „Externals“ entfernt, damit nur die eigens erstellten Klassen sichtbar bleiben. Anschliessend wurden die Tests, Implementation und Interfaces gruppiert und eingefärbt, damit das Diagramm überschaubarer wird (siehe Abbildung 32 - Übersicht Abhängigkeiten).

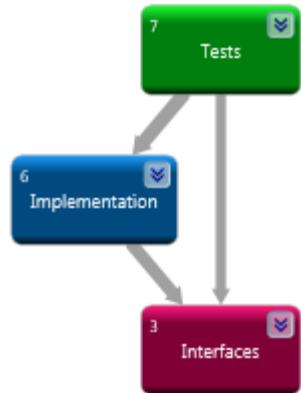


Abbildung 32 - Übersicht Abhängigkeiten

Nachfolgend sind die Klassen pro Gruppierung und weitere Details zu den Abhängigkeiten ersichtlich.

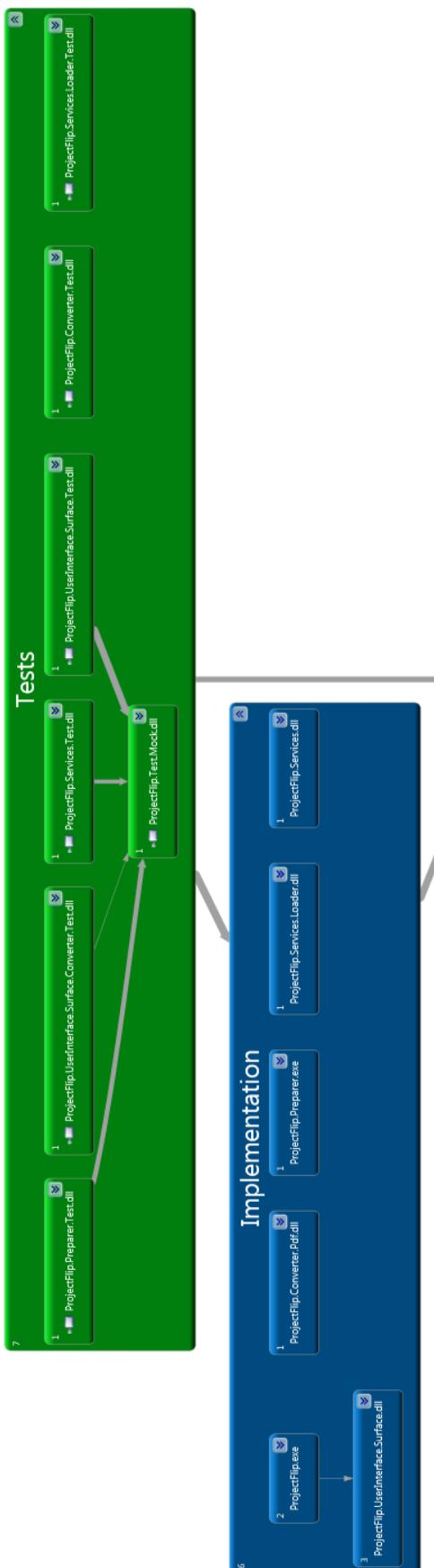


Abbildung 34 - Übersicht Abhängigkeiten mit Details

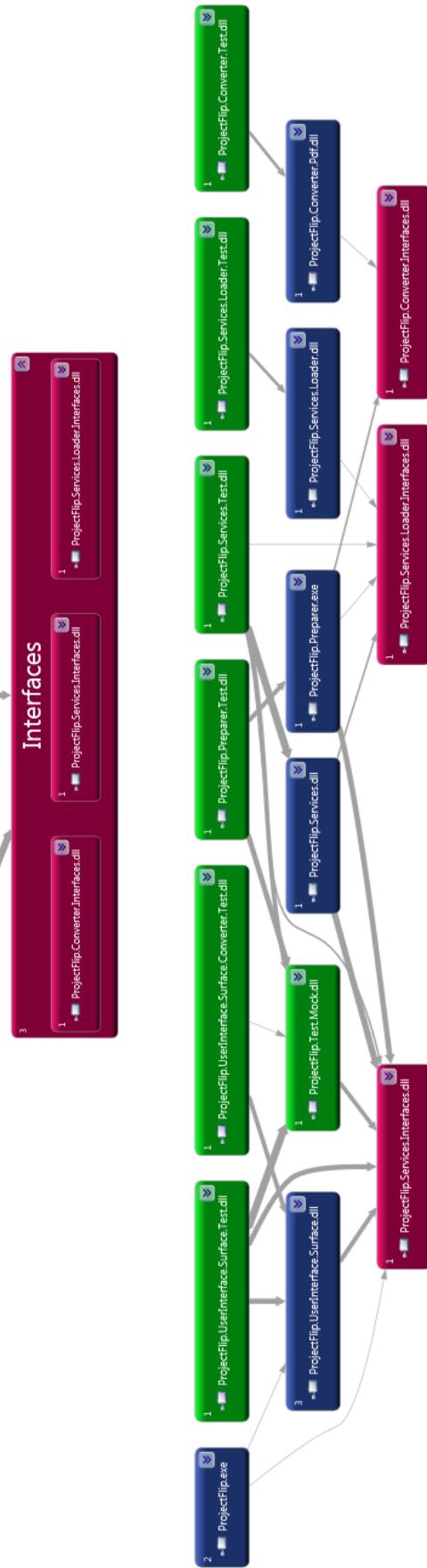


Abbildung 33 - Details Abhangigkeiten

Um die Schichtung mit einem alternativen Tool zu beweisen, wurde mit NDepend folgendes Diagramm generiert.

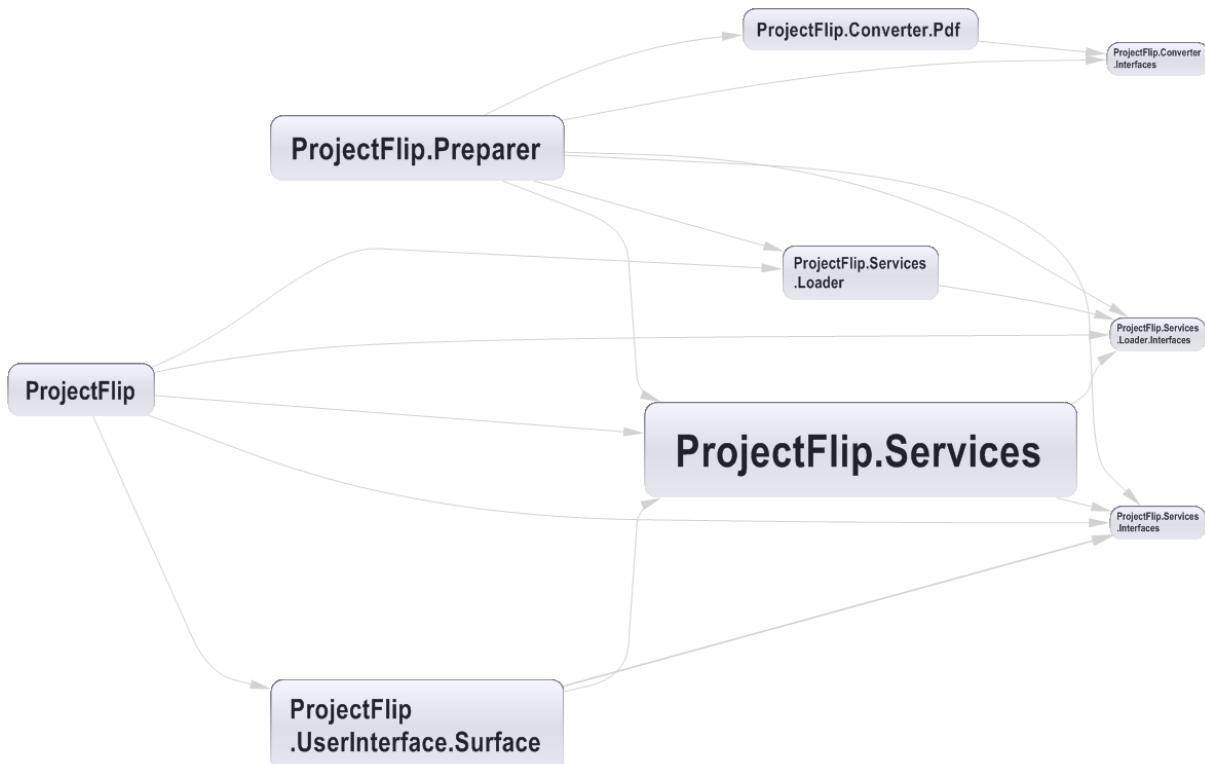


Abbildung 35 - NDepend Dependency Graph

Ganz links befindet sich das Startup Projekt, weiter rechts folgen das User Interface und der Preparer, dann die Services, Loader und Converter und ganz rechts schliesslich noch die Interfaces.

In den folgenden Unterkapiteln werden die einzelnen Projekte der Visual Studio Solution genauer beschrieben. Die Test Projekte jedoch werden nicht weiter dokumentiert, da ihre Namen ihren Zweck schon genügend vermitteln.

V.6.3.3.1 Interfaces

Um die Implementation untereinander zu entkoppeln, wurden Interfaces eingesetzt. Die Abbildung 36 - Übersicht Interfaces gibt einen Überblick.

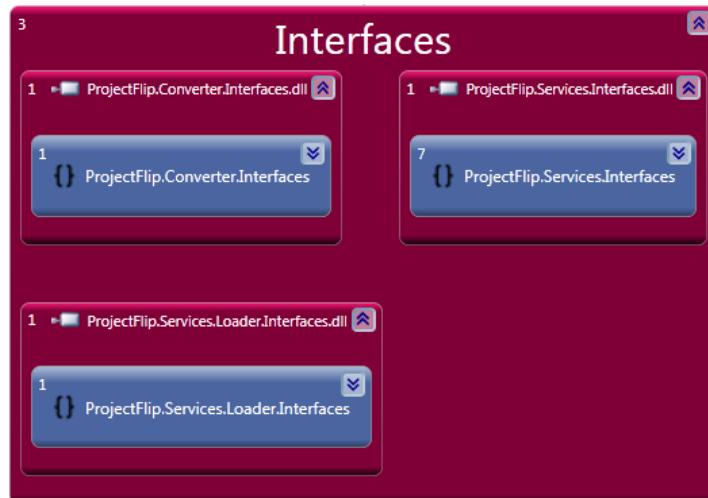


Abbildung 36 - Übersicht Interfaces

Die Interfaces werden nachfolgend genauer beschrieben.

V.6.3.3.1.1 Converter.Interfaces

Das Interface `IConverter` wird für die Konvertierung vom PDF Dokument in ein XPS Dokument eingesetzt. Das ermöglicht ein einfaches Austauschen der Implementation. Dies wird in Zukunft, sobald mehr solche Konverter gratis zur Verfügung stehen und sich das XPS Dateiformat mehr durchgesetzt hat, auch gut möglich sein.

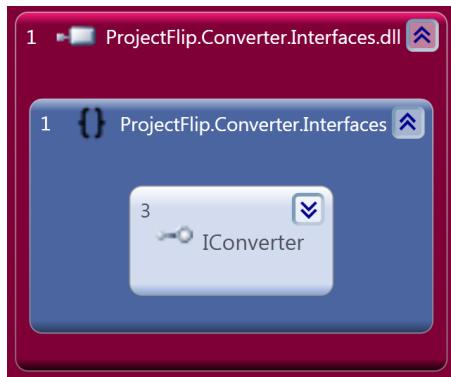


Abbildung 37 - Converter.Interfaces

V.6.3.3.1.2 Services.Interfaces

Die Interfaces `IAggregator`, `IMetadata` und `IMetadataType` repräsentieren die Metadaten der Project Notes. So wird als Beispiel der Begriff „C++“ als `IMetadata` und die Kategorie „Technologie“ als `IMetadataType` gespeichert. Der `IAggregator` ist für das Mapping verantwortlich, mit Hilfe dessen beispielsweise die Bezeichnungen „JavaBeans“ und „Java EE“ zum Begriff „Java“ zusammengefasst werden können.

Der `ICultureHelper` ist dafür verantwortlich, dass eine Sprache registriert werden kann. Dies wird zur Anzeige eines XPS benötigt.

Die `ICyclicCollectionView<T>` ist eine zyklische Liste, auf der gefiltert werden kann. Sie implementiert das `INotifyPropertyChanged`, damit sie die involvierten Komponenten benachrichtigen kann.

Die `IProjectNote` stellt eine ProjectNote dar. Neben den Properties (z.B. Title) stellt sie auch eine Preload-Methode zur Verfügung, um das XPS Dokument mittels Eager Loading vorzuladen.

Der `IProjectNotesService` stellt den Service für die Project Notes zur Verfügung. Es bietet zwei Property Getters – einer für die `IProjectNotes` und einer für die `IMetadatas`.

Ein Teammitglied wird durch die `IPerson` repräsentiert. Der `IGravatarService` stellt den Service für das Team dar, also die einzelnen `IPersons`.

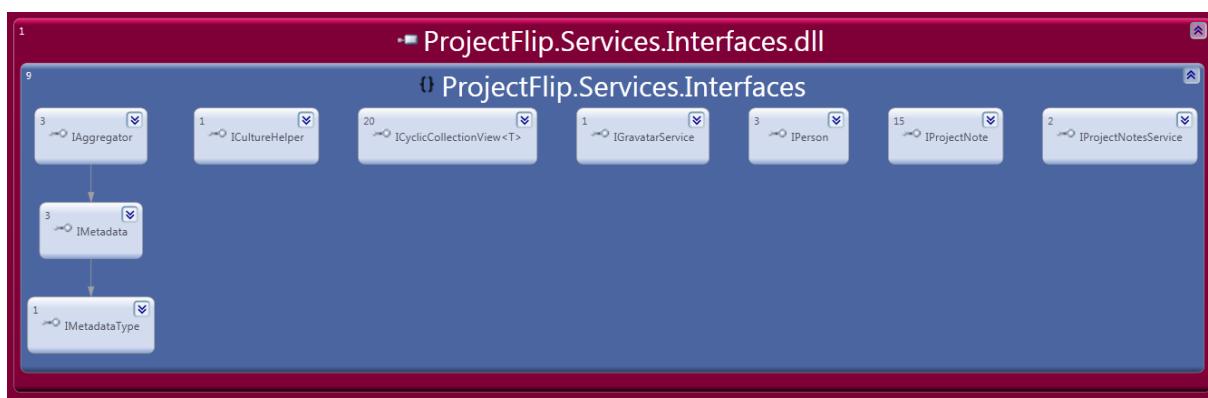


Abbildung 38 - Services.Interfaces

V.6.3.3.1.3 Loader.Interfaces

Der IProjectNotesLoader stellt die Möglichkeit, IProjectNotes zu laden, zur Verfügung. Das Interface ist deshalb in einem separaten Assembly, damit der Preparer und die eigentliche Applikation dieses gemeinsame Interface implementieren können.

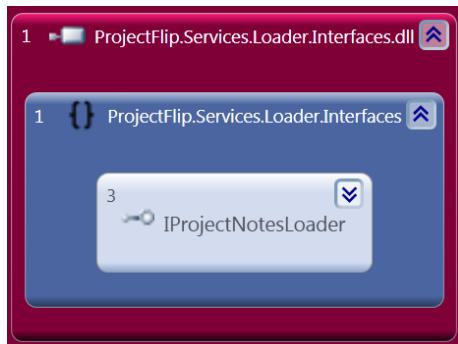


Abbildung 39 - Loader.Interfaces

V.6.3.3.2 Implementation

Die Implementation ist von den Interfaces abhängig. Dadurch können die Implementationsen voneinander entkoppelt werden. Dies ist in Abbildung 40 - Implementation Übersicht ersichtlich.

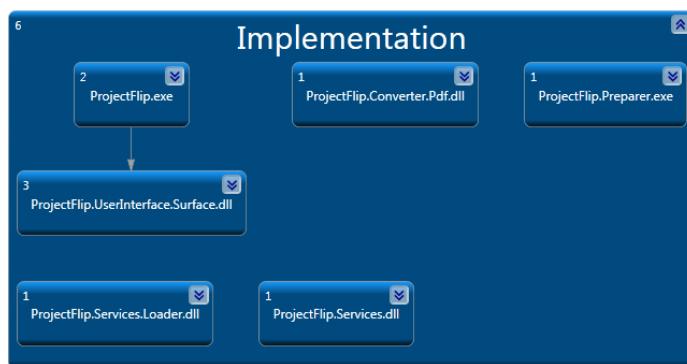


Abbildung 40 - Implementation Übersicht

V.6.3.3.2.1 Services.Loader

Der ProjectNotesLoader kann eine Liste von IProjectNotes laden. In der aktuellen Version werden die Daten von einer Tab-getrennten Textdatei gelesen, in Zukunft können diese aber auch von einer Datenbank oder vom Sharepoint geladen werden.

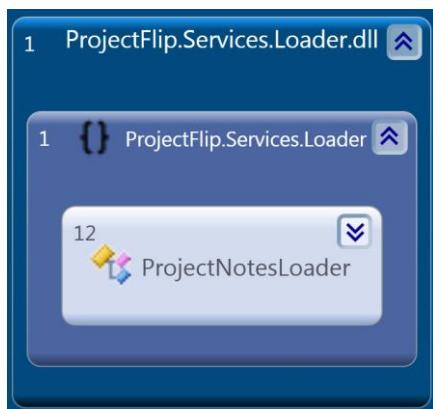


Abbildung 41 - Services.Loader

V.6.3.3.2.2 Converter.Pdf

Der PdfConverter ist dafür verantwortlich, das PDF Dokument in ein XPS Dokument zu konvertieren. Da die Konvertierung mithilfe des Adobe Readers geschieht, kann die AcrobatLocation im Settings File konfiguriert werden. Standardmäßig ist der Pfad auf „C:\Program Files (x86)\Adobe\Reader 10.0\Reader\AcroRd32.exe“ gesetzt.

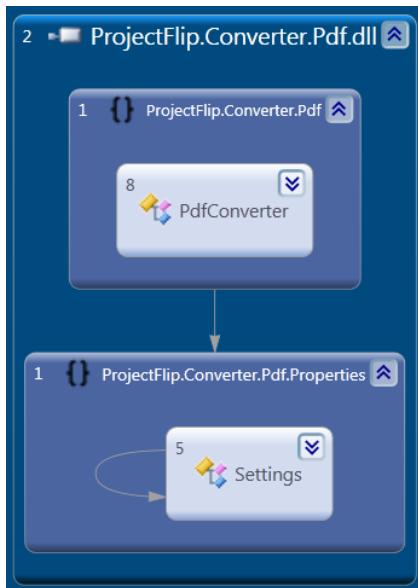


Abbildung 42 - Converter.Pdf

V.6.3.3.2.3 Preparer

Der Preparer ist dafür verantwortlich, dass die Project Notes der Applikation im entsprechenden Format zur Verfügung steht. Konkret heisst das: pro Project Note ein XPS Dokument und ein Bild (243x243 Pixel gross). Dazu verwendet der Preparer einen IProjectNotesLoader und einen IConverter.

Da der Preparer vor der Applikation und unabhängig ausgeführt wird, ist er als Executable (exe) kompiliert.

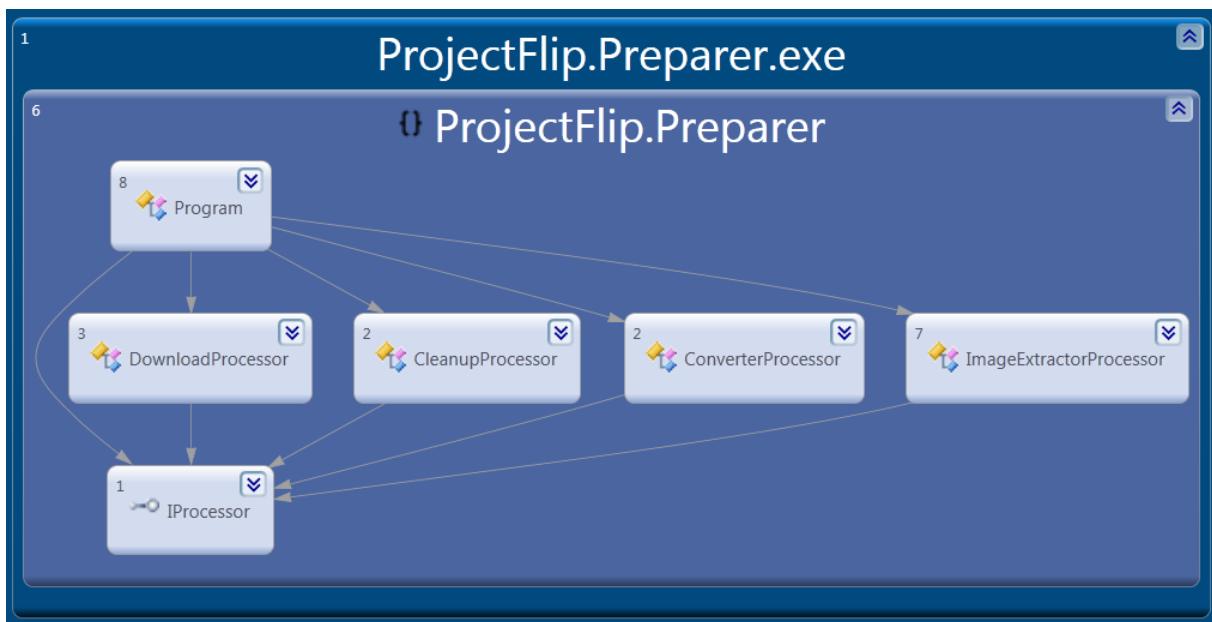


Abbildung 43 - Preparer

Beim Preparer sind die einzelnen IProcessors wichtig. Als erstes wird der DownloadProcessor verwendet, der das PDF Dokument herunterlädt und lokal speichert. In einem zweiten Schritt wird der ConverterProcessor verwendet, der das PDF Dokument in ein XPS Dokument konvertiert. Der ImageExtractorProcessor extrahiert schliesslich noch ein Bild aus dem XPS Dokument. Tritt dabei ein Fehler auf und befinden sich die Dateien dadurch in einem fehlerhaften Zustand, so werden diese vom CleanupProcessor entfernt.

Um die Ausführung zu beschleunigen, wurde die Verarbeitung der einzelnen Project Notes parallelisiert. Das folgende Sequenzdiagramm zeigt den Ablauf auf (in der Liste „processors“ sind Instanzen der einzelnen IProcessors enthalten):

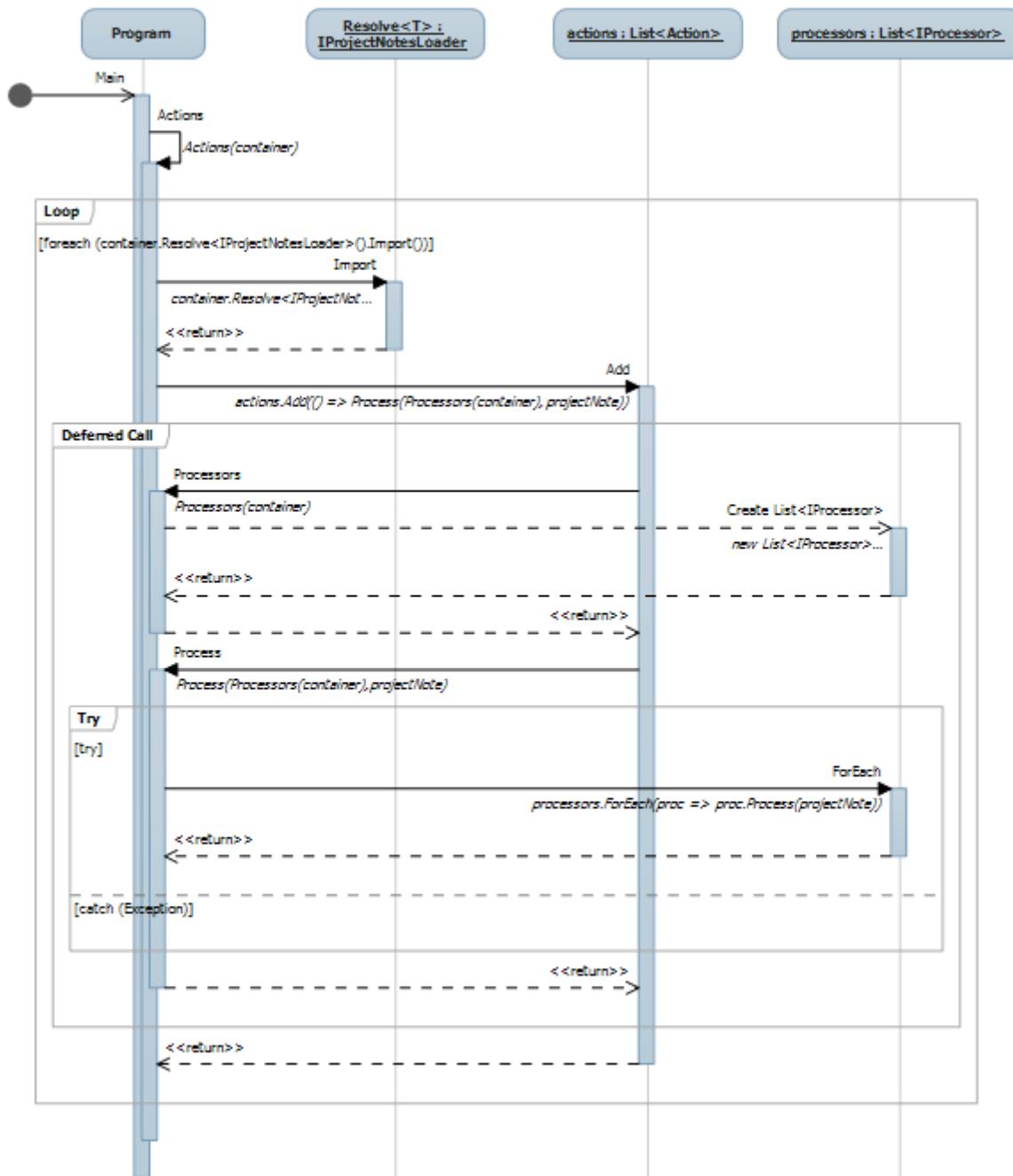


Abbildung 44 - Sequenzdiagramm Preparer

V.6.3.3.2.4 Services

Der Service stellt die IProjectNotes und die aggregierten IMetadata Objekte zur Verfügung. Er erhält zuerst die Rohdaten vom IProjectNotesLoader und baut dann aufgrund der Eingaben die Objektstruktur mit den IProjectNotes und IMetadatas auf. Die Metadaten werden beim Import sogleich aggregiert. Um auch die Personen, die am Projekt mitgearbeitet haben, darzustellen, stellt der GravatarService eine Liste von Persons zur Verfügung. Weiter implementiert das Service Projekt bestimmte Hilfsklassen, wie zum Beispiel den MetadataComparer.

Damit nach Project Notes gefiltert werden kann, stellt der Service eine CyclicCollectionView<T> zur Verfügung. Diese implementiert das Interface ICyclicCollectionView<T>.

Der Service kann auch konfiguriert werden, so gibt es eine Einstellung ProjectNotesUrl, die zurzeit auf die HTTP URL von der Zühlke Engineering AG konfiguriert ist: „http://www.zuehlke.com/uploads/tx_zepublications/“. Von diesem Verzeichnis aus werden die Project Notes heruntergeladen.

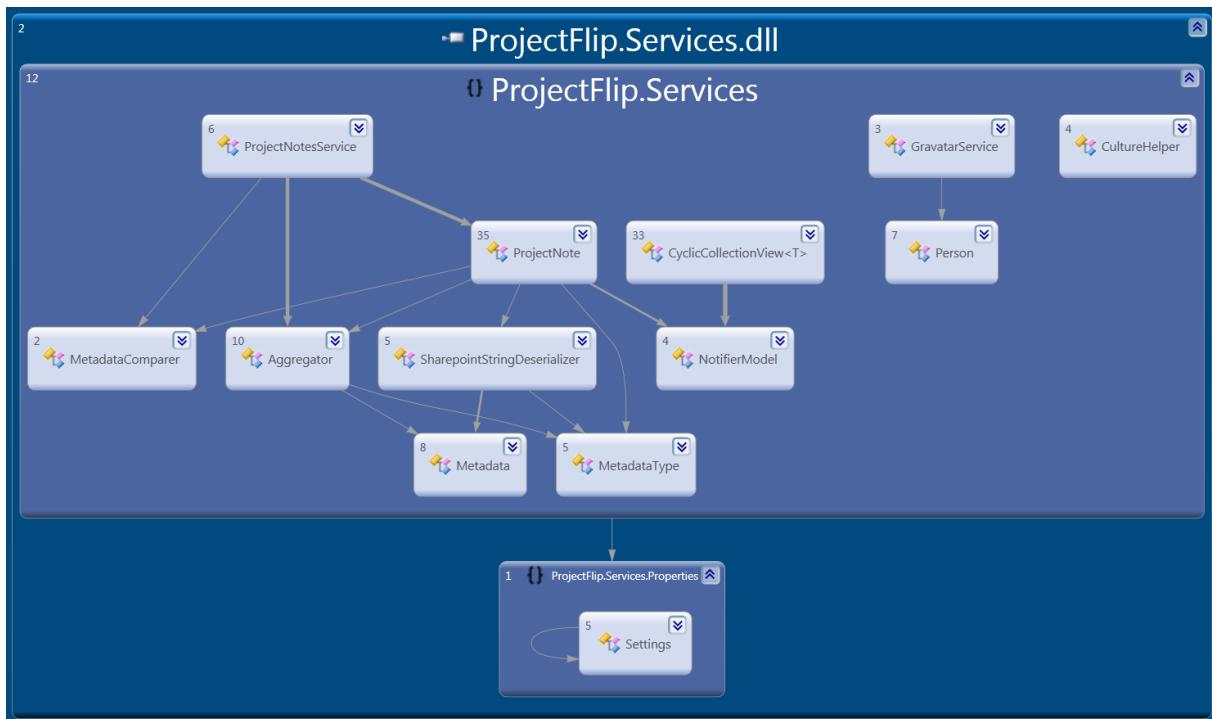


Abbildung 45 - Services

V.6.3.3.2.5 UserInterface.Surface

Das UserInterface.Surface Assembly ist für das GUI verantwortlich. Es benutzt zur Entkopplung der View und des Models das MVVM Pattern (siehe Unterkapitel V.6.4.1 MVVM).

Das OverviewWindowViewModel ist dafür verantwortlich, die Commands der View entgegenzunehmen, eine Aktion auszuführen (Bsp.: einen Filter setzen) und dann die View zu benachrichtigen (mithilfe von INotifyPropertyChanged). Das GravatarsViewModel verwaltet die Personen und stellt die Schnittstelle zwischen dem IGravatarService und dem Gravatars User Control dar.

Die Klasse Command implementiert das Interface ICommand, welches vom OverviewWindowViewModel benutzt wird.

Das ScrollToTopBehaviour wird benötigt, damit bei einer Änderung der Elemente im Container automatisch nach oben gescrollt wird.

Das OverviewWindow stellt das Hauptfenster dar. Die Klassen AboutView, DetailView, FilterDetailView, FilterView, Gravatars, ZoomInIcon und ZoomOutIcon stellen einzelne UserControls dar, die vom OverviewWindow benutzt werden.

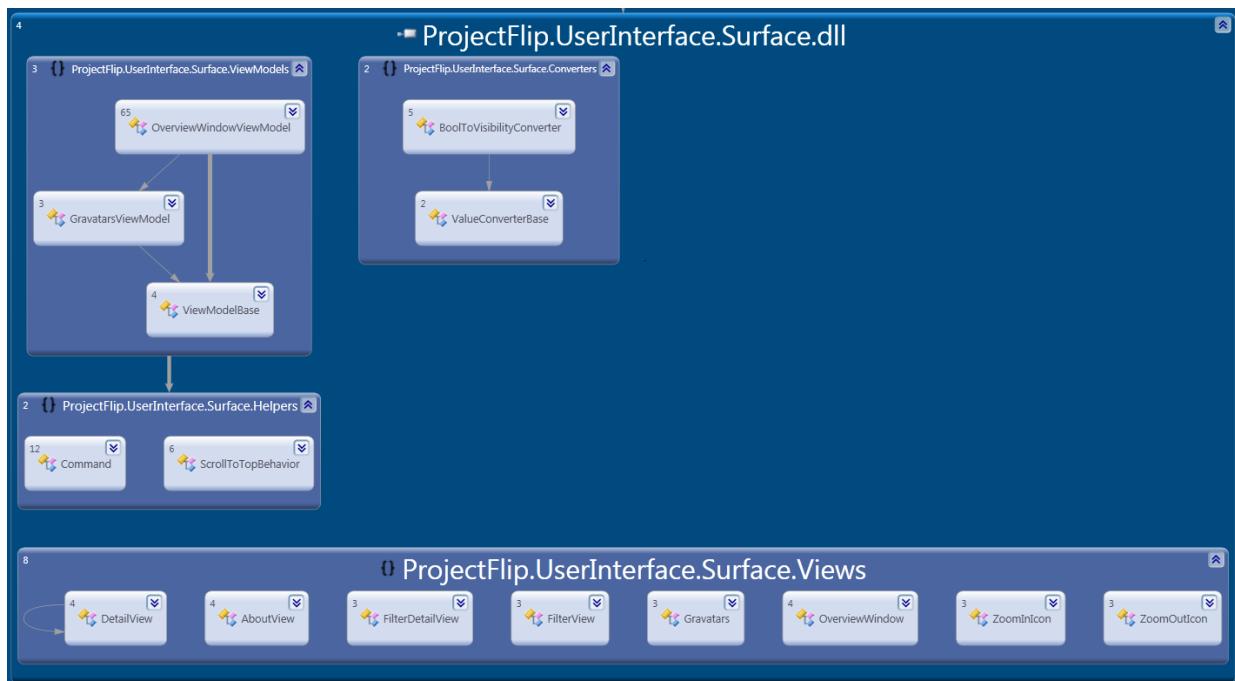


Abbildung 46 - UserInterface.Surface

V.6.3.3.2.6 ProjectFlip

Bei diesem Projekt handelt es sich das Startup Projekt. Es ist verantwortlich für die Zusammenfügung der richtigen Komponenten und schliesslich für das Starten des GUIs. Hier wurde Dependency Injection eingesetzt, und zwar mittels Microsoft Unity (siehe Unterkapitel V.6.4.3 Dependency Injection mit Microsoft Unity). So kann an einem zentralen Ort konfiguriert werden, welche Komponenten instanziert werden sollen.

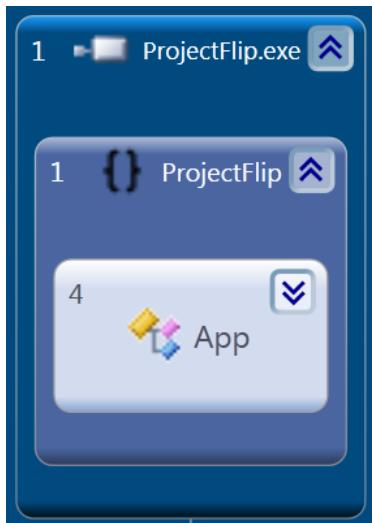


Abbildung 47 - ProjectFlip

V.6.3.4 UI Design

V.6.3.4.1 Allgemein

Für diese Applikation wurde WPF verwendet. Um die View vom Model zu trennen, wurde MVVM (siehe Unterkapitel V.6.4.1 MVVM) eingesetzt. Durch die Verwendung dieses Patterns konnte fast vollständig auf Code Behind verzichtet werden, wodurch es möglich war, die Logik der View im ViewModel über Unit Tests zu prüfen.

V.6.3.4.2 Unterteilung der Views

In vielen Applikationen kann zwischen verschiedenen Ansichten gewechselt werden. Bei Project Flip 2.0 ist dies jedoch so geregelt, dass der Startscreen immer im Hintergrund ersichtlich bleiben muss. Die anderen Komponenten (Detailansicht, Filter View, Info View) werden überlagert und mit einem halbtransparenten, abgedunkelten Hintergrund dargestellt.

V.6.3.4.2.1 Overview

Die Übersicht ist in zwei Teile eingeteilt: den Filter und die Listenansicht.

312 von 312 Projekten

Auswahlkriterien

Kriterien hinzufügen

i

Einweg Lebensmittelverpacku...

Entwickelt für den Weltmarkt: yource-Ma...

Sicherer Web-Aufritt dank regelmä... Sicherheit

Web-Applikation mit eingegebauter Sicherheit

Die Scannung mit der Maus

Sicherheit für neue biometrische Datenpla...

Optimierte Prozesse und neue Software bringe...

Haarsträubende Ideen

Innovative Regelungstechnik sen...

Energie sparen beim Duschen - wie groß is...

Mehr Effizienz durch Turbolader einfach vis...

Fit für besseres Hören mit neuer Software-Pl...

Basler Privatbank profitiert von individu...

Gut informierte Kunden fahren lieber Bahn

Scrum ebnet den Weg zur neuen Maschine

70% tiefere Kosten für IT-Infrastruktur-Betrieb

Projektmanagement beim Umzug der Flugs...

End 2 End Transparenz dank Webservice Barc...

Abbildung 48 - Übersicht

Auf der linken Seite befindet sich der Filter. Auf der rechten Seite die Project Notes aufgelistet werden.

V.6.3.4.2.2 Filter View

Wird in der Übersicht auf den Knopf „Kriterien hinzufügen“ gedrückt, so öffnet sich die Filter View.

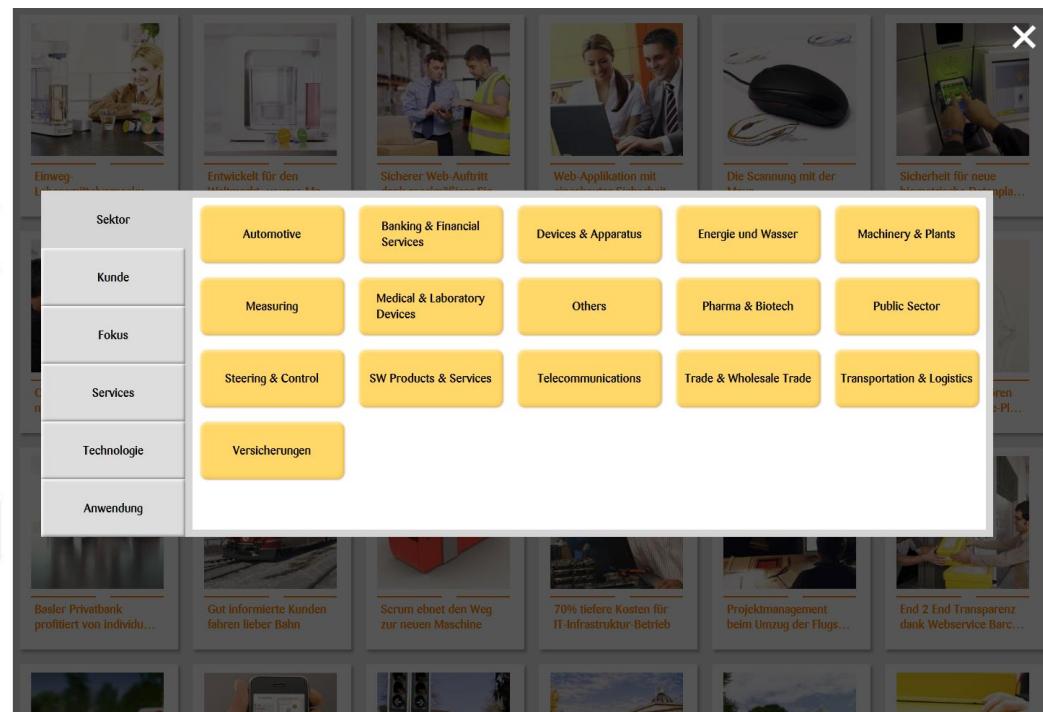


Abbildung 49 - Filter View

Wird nun ein Filterkriterium durch Antippen hinzugefügt, so wird dieses Kriterium in den Filter eingefügt, die Liste der Project Notes wird entsprechend gefiltert und die Ansicht wechselt zurück zur Übersicht (siehe Abbildung 50 - Gefilterte Übersicht).

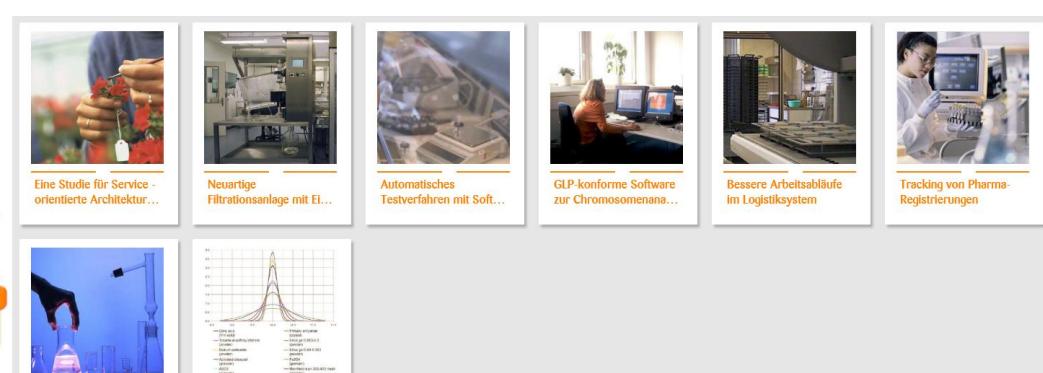


Abbildung 50 - Gefilterte Übersicht

V.6.3.4.2.3 Info View

Wird in der Übersicht auf den Infoknopf gedrückt, so öffnet sich die Info View.

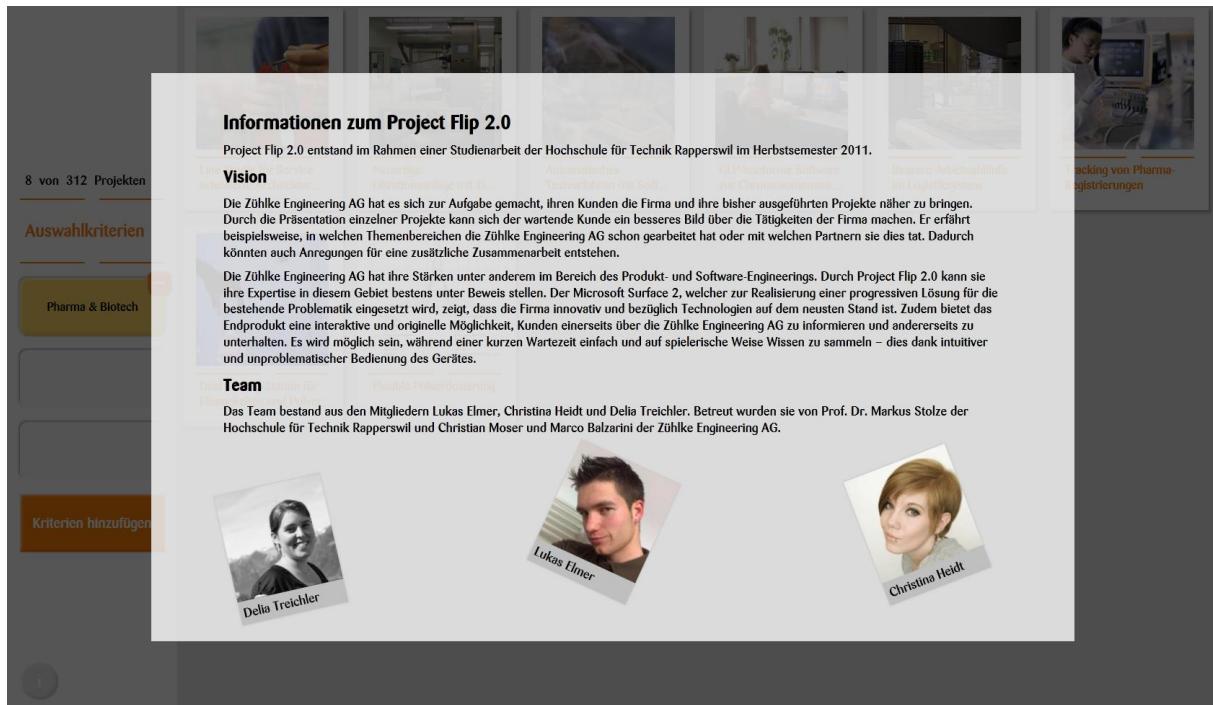


Abbildung 51 - Info View

V.6.3.4.2.4 Detail View

Wird in der Übersicht auf eine Project Note gedrückt, so öffnet sich diese in der Detailansicht.

Projektdaten

- Sektor: Pharma & Biotech
- Kunde: Syngenta
- Fokus: Software Solutions
- Services
- Integration
- Service-oriented Architectures
- Technologie: .NET
- BizTalk
- Eclipse

Project Note

Eine Studie für Service-orientierte Architektur (SOA)

Der Proof-of-Concept für die SOA (Service-orientierte Architektur) wird einer Machbarkeitsstudie unterzogen. Diese stützt sich auf verschiedene Prototypen und zeigt, was technisch machbar und wirtschaftlich sinnvoll ist.

Zunächst erarbeitet das Projektteam zusammen mit dem Kunden die Anforderungen an eine SOA-Implementierung und dokumentiert diese. Die SOA soll mit Web Services und den folgenden Quality-of-Services realisiert werden: Single Sign-On (SSO) mit Kerberos, Reliable Messaging, Long-running Transactions und Compensation. Services sollen auch ohne traditionelle Programmierung mit der Business Process Execution Language (BPEL) zu einem Geschäftsprozess „orchestriert“ werden können. Die Spezialisten der Kunden möchten die neue Entwicklungsplattformen .NET und Java auch weiterhin nutzen. Ein Vorteile ist die Nutzung von Services nutzen. Sie stellen mit unterschiedlichen Anwendungen die Anforderungen und überprüfen die Funktionen. Ein Vorteile ist die, wie in einem kurzen Zeitraum und mit wenigen Ressourcen die benötigten Empfehlungen in einer übersichtlichen und leicht verständlichen Form zu erhalten.

Kundennutzen

- Das breite Know-how der Zühlke Mitarbeiter/-innen erlaubt es, alle Kundenanforderungen optimal und ausgewogen zu erfüllen. Ein Vorteile ist die, dass die Zühlke Mitarbeiter/-innen erfahren, was technisch möglich ist und wo die Grenzen liegen.
- Technische Know-how der Zühlke ergibt die Erfahrung.
- Bank reicher Auswahl von Produkten sind die Ergebnisse objektiv und erlauben langfristig tragbare Empfehlungen.

Technische Daten

Plattformen: .NET Framework auf Windows, JBoss auf Linux und Windows
.NET Entwicklung: Visual Studio.NET, WSE
Java Entwicklung: Eclipse /MyEclipse Plug-in, JWSDP
Integrationstools: MS BizTalk Server, Oracle BPEL Process Manager
Datenbank: Oracle Datenbank

Abbildung 52 - Detail View

Aus den Projektdaten kann ebenfalls ein Filterkriterium ausgewählt und dem Filter hinzugefügt werden. Mit den Pfeilen kann zur nächsten (nach rechts) beziehungsweise vorherigen (nach links) Project Note navigiert werden. Außerdem kann, um den Text besser lesen zu können, die Project Note durch Antippen der Lupe vergrößert werden (siehe Abbildung 53 - Detail View mit Zoom).

Projektdaten

- Sektor: Pharma & Biotech
- Kunde: Syngenta
- Fokus: Software Solutions
- Services
- Integration
- Service-oriented Architectures
- Technologie: .NET
- BizTalk
- Eclipse

Project Note

Eine Studie für Service-orientierte Architektur (SOA)

Der Proof-of-Concept für die SOA (Service-orientierte Architektur) wird einer Machbarkeitsstudie unterzogen. Diese stützt sich auf verschiedene Prototypen und zeigt, was technisch machbar und wirtschaftlich sinnvoll ist.

Aufgabe

Zunächst erarbeitet das Projektteam zusammen mit dem Kunden die Anforderungen an eine SOA-Implementierung und dokumentiert diese. Die SOA soll mit Web Services und den folgenden Quality-of-Services realisiert werden: Single Sign-On (SSO) mit Kerberos, Reliable Messaging, Long-running Transactions und Compensation. Services sollen auch ohne traditionelle Programmierung mit der Business Process Execution Language (BPEL) zu einem Geschäftsprozess „orchestriert“ werden können.

Umsetzung

Zunächst erarbeitet das Projektteam zusammen mit dem Kunden die Anforderungen an eine SOA-Implementierung und dokumentiert diese. Die SOA soll mit Web Services und den folgenden Quality-of-Services realisiert werden: Single Sign-On (SSO) mit Kerberos, Reliable Messaging, Long-running Transactions und Compensation. Services sollen auch ohne traditionelle Programmierung mit der Business Process Execution Language (BPEL) zu einem Geschäftsprozess „orchestriert“ werden können.

Technische Daten

Plattformen: .NET Framework auf Windows, JBoss auf Linux und Windows
.NET Entwicklung: Visual Studio.NET, WSE
Java Entwicklung: Eclipse /MyEclipse Plug-in, JWSDP
Integrationstools: MS BizTalk Server, Oracle BPEL Process Manager

Abbildung 53 - Detail View mit Zoom

V.6.3.5 Prozesse und Threads

Die drei Tiers (Webserver, Sharepoint Server, Surface 2) agieren unabhängig. Da weder am Webserver noch am Sharepoint Server Anpassungen vorgenommen wurden, werden diese zwei Tiers nicht weiter beschrieben. Stattdessen wird auf den Surface 2 genauer eingegangen.

Als Grundmodell wird das .NET Framework mit WPF verwendet. Nachfolgend werden einige Spezialfälle, die in der Applikation programmiert wurden, genauer betrachtet.

V.6.3.5.1 Konversation PDF zu XPS

Um das PDF in ein XPS zu konvertieren, wird der Adobe Reader und der XPS Dokument Writer (Drucker) benutzt. So muss für den Adobe Reader ein eigener Prozess gestartet werden, der dann nicht weiter kontrolliert werden kann aufgrund einer sauberer Schnittstellendefinition des Adobe Readers. Deshalb kann in der Konfiguration die Wartezeit („SecondsToWait“), bis der Adobe Prozess beendet wird, angegeben werden. Standardmäßig ist dieser Wert mit 30 belegt, das heißt, der Adobe Reader hat maximal 30 Sekunden Zeit, bis er beendet wird.

V.6.3.5.2 Parallel Preparer

Beim Preparer, also der Komponente, die für das Vorbereiten (Download PDF, Konvertierung PDF in XPS, Extraktion eines Bildes) der Project Notes verantwortlich ist, ist es sinnvoll, die Verarbeitung zu parallelisieren. Es wird für jede Project Note ein eigener Thread gestartet, da die Verarbeitung pro Project Note seriell erfolgen muss. Diese gestarteten Threads können dann unabhängig arbeiten. Wegen der grossen Verzögerung, welche beim Download oder beim Lesen von der Festplatte entstehen, ist eine parallele Verarbeitung performanter.

Vorteilhaft ist, dass die einzelnen Project Notes völlig unabhängig voneinander sind und deshalb nicht speziell synchronisiert werden müssen. Außerdem wird der interne ThreadPool von .NET benutzt, welcher weitere Performancevorteile mit sich bringt.

V.6.3.5.3 Asynchrones und verzögertes Laden des XPS Dokuments

In der Detailansicht wird das XPS Dokument angezeigt. Zu Beginn des Projektes wurde das XPS Dokument sofort geladen, sobald die Detailansicht geöffnet wurde. Dies führte dazu, dass der PC dem Laden des XPS Dokuments mehr Priorität gab als der Animation, die beim Öffnen der Detailansicht ausgeführt werden sollte. Daher war diese Animation kaum sichtbar.

Aus diesem Grund wird das Laden des XPS Dokuments erst nach Ablauf der Animation ausgeführt, also um etwa 500ms verzögert. Eine mögliche Umsetzung der Ladeverzögerung wäre das Erstellen eines neuen Threads, welcher zuerst 500ms schlafend und danach das XPS Dokument lädt. Nun stellt sich aber das Problem, dass nur der GUI Thread das XPS Dokument laden und das Property im ViewModel setzen kann. Diese Einschränkung besteht, weil WPF nicht multithreading safe ist.

Deshalb wurde von der Betreuungsperson der Zühlke Engineering AG, Christian Moser, empfohlen, den Dispatcher mit einem Timer zu benutzen, um so das Dokument verzögert zu laden (siehe VII Anhang, C Sitzungsprotokoll). Zusätzlich wird eine Synchronisation benötigt, welche mit einem Lock auf das aktuelle Objekt realisiert wird.

V.6.4 Patterns

Die drei wichtigsten Patterns, die angewendet wurden, sind MVVM, Flyweight und Dependency Injection mit Microsoft Unity.

V.6.4.1 MVVM

Das MV⁴VM („Model“, „View“, „ViewModel“) Pattern wird benötigt, um die View vom Model zu entkoppeln. Deshalb wird als Zwischenglied ein ViewModel erzeugt, das die Commands des GUIs abarbeitet und die verfügbaren Elemente dem GUI zur Verfügung stellt.

Die grundlegende Idee dahinter ist, dass sich das GUI schneller ändert als die Businesslogik und deshalb die zwei Komponenten möglichst stark abzutrennen sind. Zusätzlich kann das das ViewModel dadurch mit Unit Tests geprüft werden.

V.6.4.2 Flyweight

Flyweight [eilbrecht07, Seite 68] wird für die Metadaten eingesetzt. Die Idee dabei ist, dass viele verschiedene Objekte die gleichen Objekte benötigen. Zum Beispiel benötigen mehrere Project Notes ein Metatag mit dem Namen „Java“. Damit dieses Objekt nicht immer wieder neu erstellt wird, referenzieren die verschiedene Project Notes das gleiche Metatag-Objekt.

Durch die Nutzung von Flyweight werden das Starten und der Betrieb der Applikation performanter, da einzelne Objekte wieder verwendet werden und nicht immer wieder neu erstellt werden müssen. Zusätzlich wird auch die Suche effizienter.

Der Nachteil besteht jedoch darin, dass die Flyweight Objekte immutable Objects, also unveränderlich, sein müssen. Deshalb verfügt das Metadata Model auch über keine Setter.

V.6.4.3 Dependency Injection mit Microsoft Unity

Damit die Komponenten jederzeit und einfach ausgetauscht werden können, wurde mit Unity Containern⁵ gearbeitet, um Dependency Injection zu ermöglichen. Für dieses Projekt mag dies ein Overhead sein, aber speziell in grösseren Projekten ist Dependency Injection sehr wichtig. Denn so können beispielsweise, auf eine einfache Art und Weise, verschiedene Software Versionen ausgeliefert werden, indem die Container ausgetauscht werden. Dependency Injection kann auch beim Testen helfen, indem man beim Unity Container die Mock Objekte registriert.

⁴ Weitere Informationen: <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>

⁵ Weitere Informationen: <http://msdn.microsoft.com/en-us/library/ff663144.aspx>

Project Flip 20

Studienarbeit

V.7 Prototype
Perspective Wall

V.7.1 Dokumentinformationen

Datum	Version	Änderung	Autor
01.12.2011	1.0	Erste Version des Dokuments	lelmer
11.12.2011	1.1	Weiter Dokumentiert, Bilder eingefügt	lelmer
13.12.2011	1.2	Korrekturen	dtreichl
20.12.2011	1.3	Fokus und Kontext ergänzt	lelmer
22.12.2011	1.4	Korrekturen	cheidt

V.7.2 Übersicht

V.7.2.1 Problem

In der Übersicht über die Project Notes muss erkannt werden können, dass nach oben oder unten gescrollt werden kann. Zusätzlich wäre es schön, wenn man die Anzahl Project Notes einfach erkennen und abschätzen könnte. Man möchte also den Fokus, ein paar wenige Project Notes, aber auch den Kontext, die ganze Liste, sehen und erkennen können.

V.7.2.2 Idee

In einer 3D-Ansicht der Project Notes kann durch den 3D-Effekt erkannt werden, dass im Hintergrund (nach oben und nach unten) noch weitere Project Notes existieren. Da die Project Notes im Hintergrund kleiner werden, kann auch die Anzahl ungefähr abgeschätzt werden. Die fokussierten Project Notes werden immer am grössten darstellt.

V.7.3 Ansatz

Um eine Lösung erarbeiten zu können, wurde zuerst ein sehr genereller Ansatz gesucht. Dieser wurde dann während mehreren Schritten verfeinert. Nachfolgend wird das Vorgehen aufgezeigt.

V.7.3.1 Genereller Ansatz

Als Grundlage wurde die Perspective Wall [chi91] genommen. Das sieht grundsätzlich folgendermassen aus:

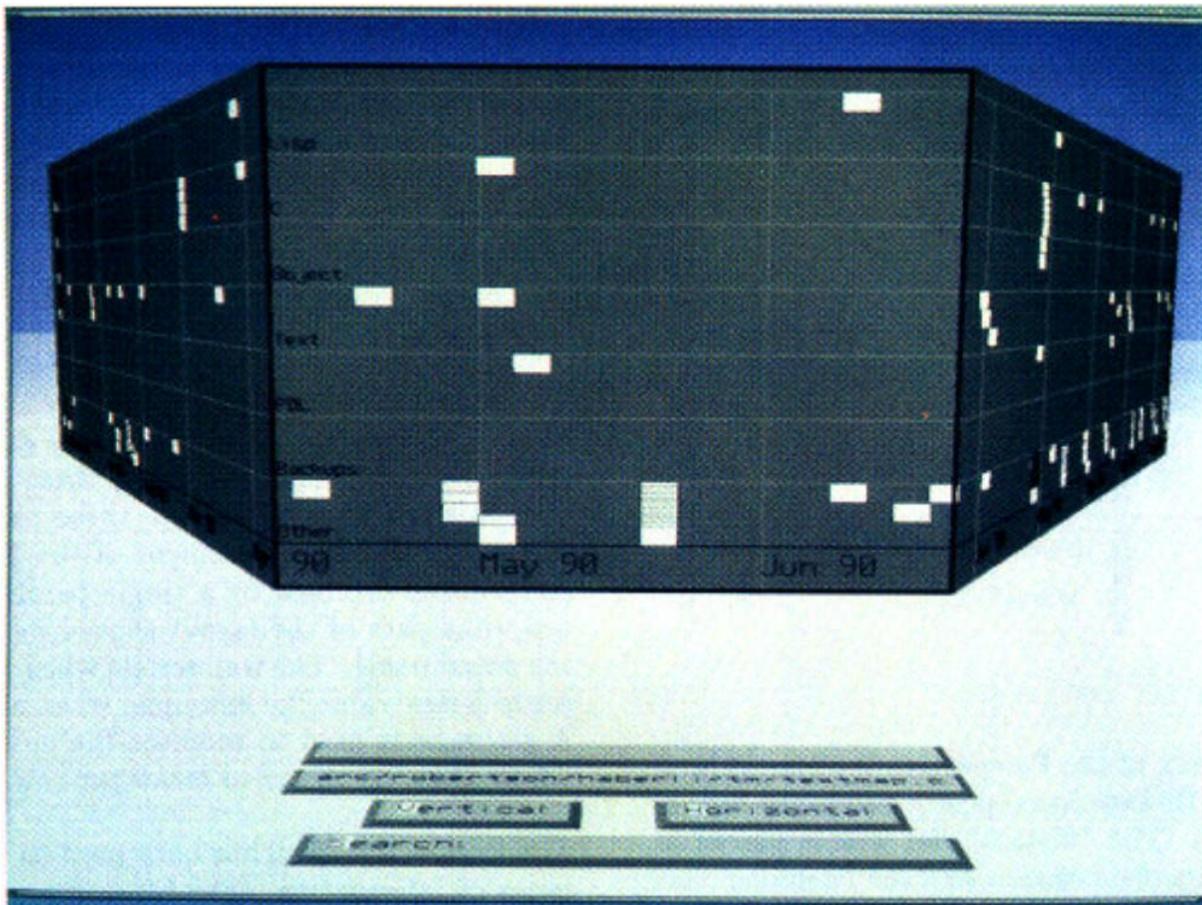


Abbildung 54 - Beispiel einer Perspective Wall, [chi91]

V.7.3.2 Ausrichtung

Da die Project Notes von oben nach unten und nicht von links nach rechts gescrollt werden sollen, wurde in einem ersten Schritt die Wall um 90 Grad gedreht. Das sieht dann folgendermassen aus:



Abbildung 55 - Skizze 3D ohne Krümmung, 90 Grad gedreht

V.7.3.3 Anzahl Elemente und Krümmungen

Die originale Perspective Wall beschränkt sich auf drei verschiedene Darstellungen: der mittlere Teil ist flach und rechteckig dargestellt, während der rechte und linke Teil der Wall perspektivisch verzogen sind.

Da die Übersicht über die Project Notes mehr als nur drei Spalten breit sein soll, wurden noch weitere Spalten hinzugefügt, und die Neigung der perspektivischen Verzerrungen angepasst. Geometrisch bedeutet das, dass die Oberfläche der abstrakten 3D-Figur von einer flachen zu einer runden Oberfläche wird. Das wiederum heisst, dass die Linien nicht mehr gerade sondern neu rund sind. Folgende Skizze zeigt ein Beispiel auf:

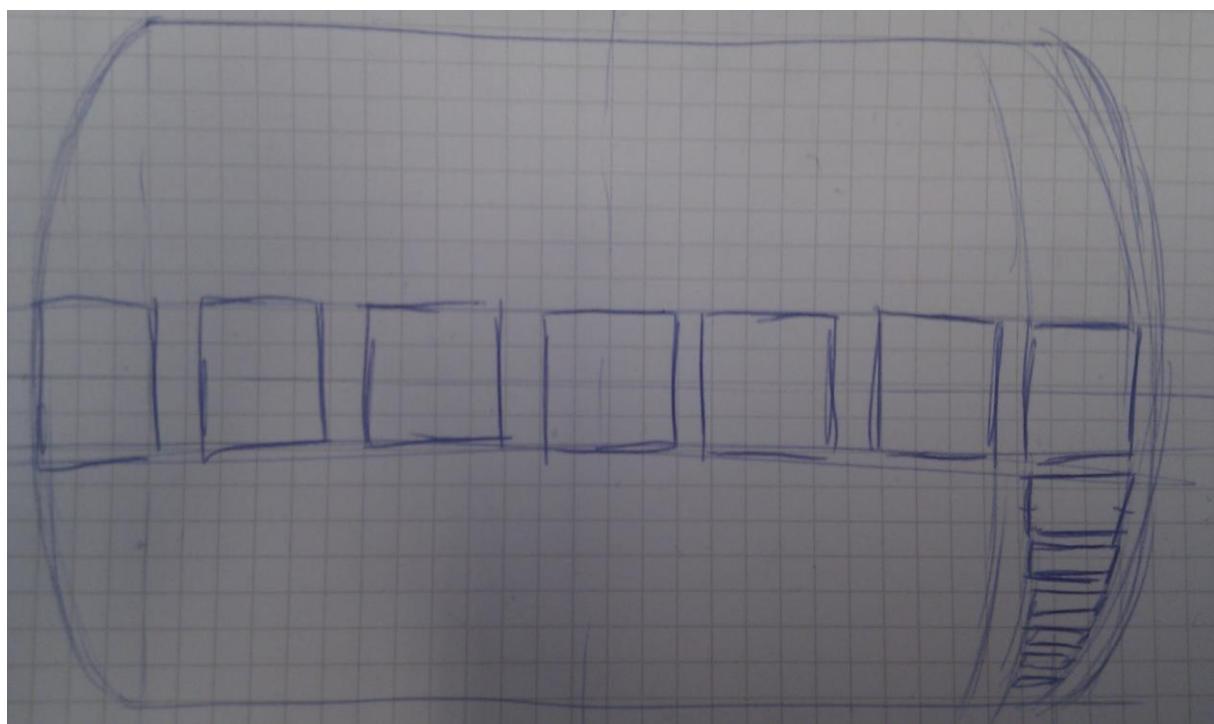


Abbildung 56 - Skizze 3D mit Krümmung

V.7.3.4 Ausblenden zu kleiner Project Notes

Damit am unteren Rand der Darstellung keine zu stark gestauchten Project Notes vorkommen, müssen einige sich unten (aus 3D-Sicht: im Hintergrund) befindende Elemente ausgeblendet werden. Dies kann z.B. so gemacht werden, dass alle Project Notes, die weniger als 5 Pixel hoch sind, ausgeblendet werden.

Dadurch wird zwar das Prinzip von Fokus und Kontext verletzt, da Elemente ausgeblendet werden. Dies ist jedoch kein echtes Problem, da sich der Benutzer durch das 3D vorstellen kann, dass sich hinter den anderen Project Notes noch weitere, kleine Project Notes befinden könnten.

V.7.4 Umsetzung

V.7.4.1 Mathematische Grundlagen

Grundsätzlich wird eine Funktion mit zwei Parametern benötigt:

- X: die Position der X-Achse
- Y: die Höhe, auf der die Project Note gezeichnet werden soll

Skizziert sieht dies wie folgt aus:

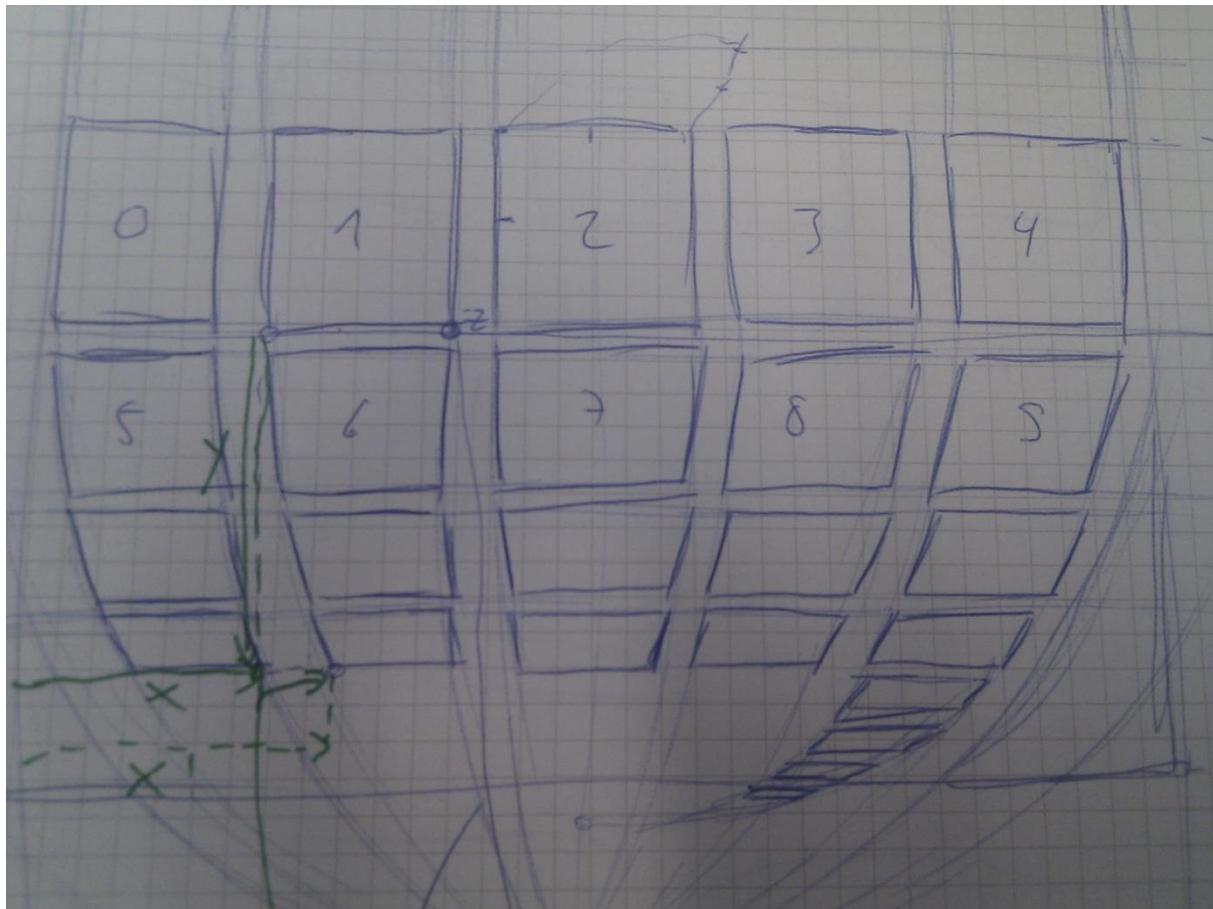


Abbildung 57 - Skizze 3D mit X, Y

Als Ergebnis wird eine Position X' , Y' zurückgegeben, also die transformierte Position der jeweiligen Project Note. Die Performance der Berechnung ist wichtig, da beim Scrollen die Berechnung jeweils für jedes einzelne Element neu durchgeführt werden muss, damit die Elemente an der richtigen Position in der richtigen Grösse und Darstellung angezeigt werden. Daher muss speziell auf das Laufzeitverhalten geachtet werden, idealerweise soll die Berechnung pro Element mit $O(1)$ erfolgen. Dies ist mithilfe von Folgen und Reihen möglich:

$$\sum_{i=0}^n k^i = \frac{k^{n+1} - 1}{k - 1}$$

Wobei

$$k \approx \frac{1}{2}$$

V.7.4.2 Programmierung

V.7.4.2.1 Implementation

Die Implementation zur Theorie stellt ein weiteres, eher schwieriges Problem dar. Da die Anforderungen klar sind, die Logik aber schwierig zu implementieren ist, wurde für die Entwicklung TDD (Test Driven Development) gewählt. Diese Tests umfassten die Berechnung der aktuellen Position, wie sich die Elemente verhalten, wenn zur nächsten Reihe gewechselt wird, wie sich die aktuelle Reihe und Spalte verhält und wie viele Elemente angezeigt werden. Als der erste Ansatz für die Implementation der 3D-Ansicht verworfen und durch einen neuen ersetzt wurde, hat sich die TDD Methode bewährt, da die zu Beginn geschriebenen Tests weiterhin verwendet werden konnten. Auch für die Optimierung der Performance war diese Vorgehensweise hilfreich.

Die Logik wurde in verschiedene Komponenten aufgeteilt:

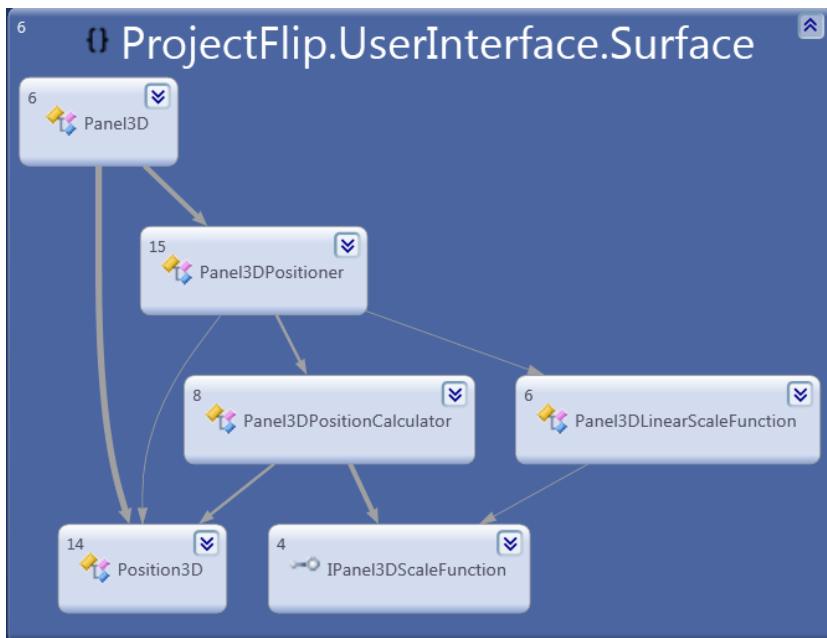


Abbildung 58 - Dependency Diagram 3D Komponenten

Position3D

Ein Value Object, das eine 3D-Position speichert.

IPanel3DScaleFunction

Stellt das Interface zur Verfügung, welches eine ScaleFunction bieten muss.

Panel3DLinearScaleFunction

Eine das Interface **IPanel3DScaleFunction** implementierende Klasse, die linear skaliert.

Panel3DPositionCalculator

Der Calculator rechnet eine bestimmte Position aus, wobei er mit einer bestimmten **IPanel3DScaleFunction** arbeitet. Als Input erhält er für jede Berechnung eine bestimmte Reihe und Spalte (X, Y Position).

Panel3DPositioner

Der Positioner ist dafür verantwortlich, die einzelnen Elemente im Panel anzugeordnen.

Panel3D

Diese Komponente stellt das Panel dar. Es ist für die Events und für den Zustand des Panels verantwortlich.

V.7.4.2.2 Resultat

Das Endergebnis ist eine komplexe 3D-Ansicht, welche in relativ wenig Zeit programmiert wurde. Die Ansicht skaliert die Reihen und Spalten für beliebige Fenstergrößen und ist performant. Die einzelnen Elemente, welche sich in der Ansicht befinden, können auch selektiert werden und auch die Effekte von WPF (z.B. Schatten, Rahmen) können ganz normal auf die Elemente angewendet werden.

Hier ein Beispiel des Schlussresultates:

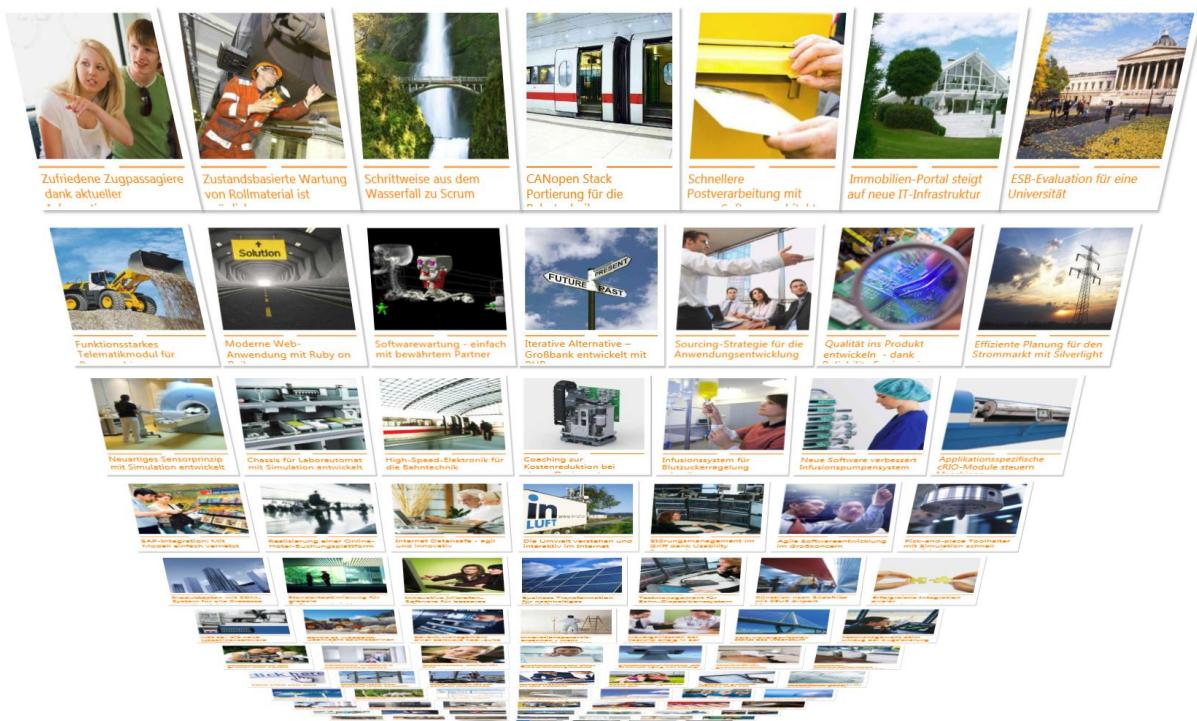


Abbildung 59 - Vollbild Ansicht

V.7.5 Weiterentwicklung

Da der Fokus dieser Arbeit auf der Grundfunktionalität der Software lag, war die Zeit zur Entwicklung von Begeisterungsfaktoren von Beginn her begrenzt. Es zeigte sich, dass die Unterstützung von interaktivem Scrolling in der Perspective Wall nicht in der verfügbaren Zeit realisierbar war. Daher wurde das 3D-Projekt eingestellt.

Falls das Scrolling noch implementiert wird, ist darauf zu achten, dass die Scroll-Auswirkung im unteren Bereich vom Panel stärker ist als im oberen. Der Grund dafür, ist, dass sich der Finger oder die Maus während dem Scrollvorgang immer über dem gleichen Element befinden soll.

Zusätzlich könnte für viele Elemente auch eine Virtualisierung der nicht sichtbaren Elemente in der Liste wichtig sein, da so die Performance gesteigert werden kann.

Weiter wäre es auch möglich, die Positions-Berechnungen zu parallelisieren, da die einzelnen Positionen unabhängig voneinander ausgerechnet werden können. Dies wird speziell in der Zukunft, wenn mehr Kerne für die Berechnung zur Verfügung stehen, wichtig sein.

V.7.6 Fazit

Die Entwicklung eines eigenen Panels ist etwas aufwändig und komplex. Wenn dazu auch noch etwas in 3D dargestellt werden muss und viel Mathematik benötigt wird, wird es umso schwieriger. Das Ganze dann auch noch skalierbar und mit guter Performance zu implementieren, vergrössert die Komplexität zusätzlich. Darüber hinaus soll der Code auch noch einfach wartbar sein.

Abschliessend muss gesagt werden, dass die Entwicklung in diesem Bereich zwar sehr anspruchsvoll war, aber auch viel Spass gemacht hat. Speziell hervorzuheben ist der Einsatz von Test Driven Development (TDD), der die Entwicklung sehr stark beschleunigt hat und durch den auch viele Fehler schnell identifiziert werden konnten.

Es ist natürlich schade, wenn eine Entwicklung nach viel Einsatz pausiert bzw. abgebrochen werden muss, aber in dieser Situation mit Zeitdruck für andere Funktionalitäten war das die einzige gangbare Lösung.

Project Flip 20

Studienarbeit

V.8 Realisierung &
Tests

V.8.1 Dokumentinformationen

Datum	Version	Änderung	Autor
28.10.2011	1.0	Erste Version des Dokuments	cheidt
31.10.2011	1.1	Unit Tests hinzugefügt	lelmer
14.11.2011	1.2	System Tests Sprint 4 hinzugefügt	cheidt
14.11.2011	1.3	Unit Tests Sprint 4 hinzugefügt	dtreichl
28.11.2011	1.4	Unit&System Tests Sprint 5 hinzugefügt	dtreichl
05.12.2011	1.5	Usability Test hinzugefügt	cheidt
11.12.2011	1.6	Test Sprint 6 hinzugefügt	lelmer
11.12.2011	1.7	Betriebs- und Installationskonzept	lelmer
12.12.2011	1.8	Code Qualität, Review	dtreichl
16.12.2011	1.9	Code Metriken, Code Reviews, Tests Sprint 7	lelmer
19.12.2011	1.10	Review	cheidt
19.12.2011	1.11	Review	dtreichl
21.12.2011	1.12	Korrekturen	cheidt

V.8.2 Tests

V.8.2.1 Unit Tests

Die untenstehenden Unit Tests wurden für im jeweiligen Sprint geschrieben und durchgeführt.

V.8.2.1.1 Sprint 3

Test Results			
	Result	Test Name	Project
<input type="checkbox"/>	Passed	LineTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed	NotifyTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed	ImportTest	ProjectFlip.Services.Loader.Test
<input type="checkbox"/>	Passed	OverviewWindowViewModelProjectNotesTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed	ImageTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed	ExtractImageTest	ProjectFlip.Preparer.Test
<input type="checkbox"/>	Passed	ConvertToListTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed	DocumentTest	ProjectFlip.Services.Test
<input checked="" type="checkbox"/>	Inconclusive	OpenNewWindowTest1	ProjectFlip.UserInterface.Surface.Test

Abbildung 60 - Unit Tests 31.10.2011

Der letzte Unit Test für das Öffnen eines neuen Fensters (Window) führte zu einigen Schwierigkeiten, die in einem Meeting mit Christian Moser, welches zu einem späteren Zeitpunkt stattfand, gelöst werden konnten. Dieser Test blieb daher zum Zeitpunkt der Testdokumentation unvollständig.

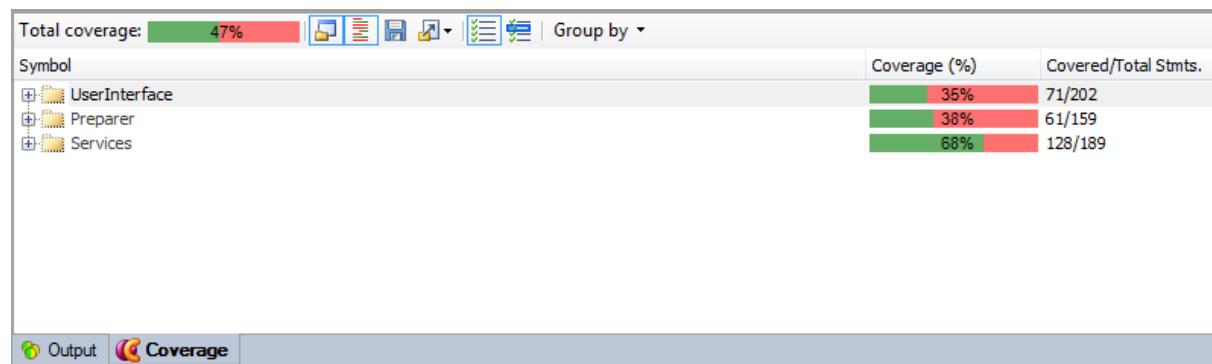
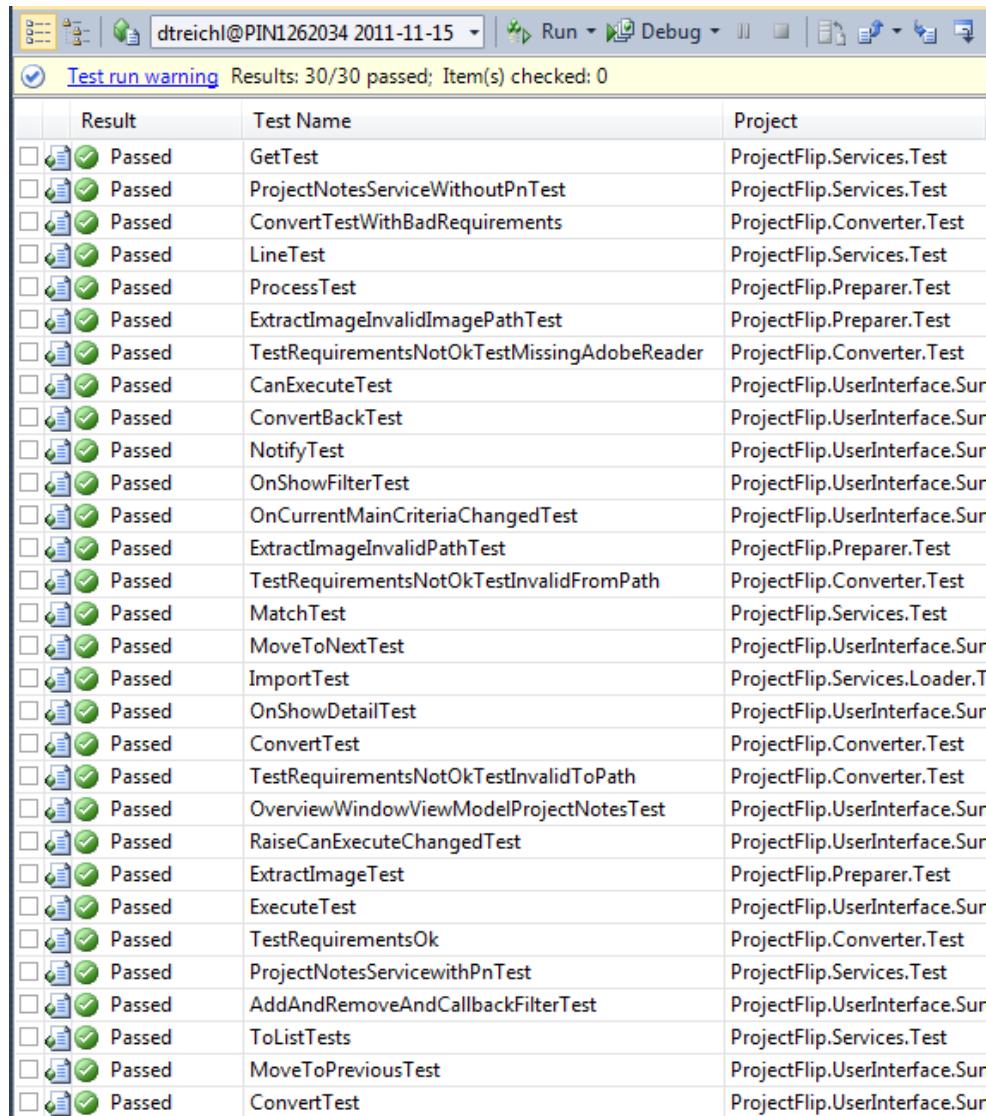


Abbildung 61 - Test Coverage 31.10.2011

Die Testabdeckung des Sprints 3 war noch nicht ideal. Ziel ist es, diese in den weiteren Sprints zu steigern.

V.8.2.1.2 Sprint 4



The screenshot shows a software interface for running unit tests. At the top, there's a toolbar with icons for Run, Debug, and other functions. Below the toolbar, a status bar displays the user's name (dtreichl@PIN1262034) and the date (2011-11-15). The main area is a table with three columns: Result, Test Name, and Project. The 'Result' column contains green checkmarks and the word 'Passed'. The 'Test Name' column lists various test methods such as GetTest, ProjectNotesServiceWithoutPnTest, ConvertTestWithBadRequirements, LineTest, ProcessTest, ExtractImageInvalidImagePathTest, TestRequirementsNotOkTestMissingAdobeReader, CanExecuteTest, ConvertBackTest, NotifyTest, OnShowFilterTest, OnCurrentMainCriteriaChangedTest, ExtractImageInvalidPathTest, TestRequirementsNotOkTestInvalidFromPath, MatchTest, MoveToNextTest, ImportTest, OnShowDetailTest, ConvertTest, TestRequirementsNotOkTestInvalidToPath, OverviewWindowViewModelProjectNotesTest, RaiseCanExecuteChangedTest, ExtractImageTest, ExecuteTest, TestRequirementsOk, ProjectNotesServiceWithPnTest, AddAndRemoveAndCallbackFilterTest, ToListTests, MoveToPreviousTest, and ConvertTest. The 'Project' column shows the names of the projects these tests belong to, such as ProjectFlip.Services.Test, ProjectFlip.Converter.Test, etc.

Result	Test Name	Project
Passed	GetTest	ProjectFlip.Services.Test
Passed	ProjectNotesServiceWithoutPnTest	ProjectFlip.Services.Test
Passed	ConvertTestWithBadRequirements	ProjectFlip.Converter.Test
Passed	LineTest	ProjectFlip.Services.Test
Passed	ProcessTest	ProjectFlip.Preparer.Test
Passed	ExtractImageInvalidImagePathTest	ProjectFlip.Preparer.Test
Passed	TestRequirementsNotOkTestMissingAdobeReader	ProjectFlip.Converter.Test
Passed	CanExecuteTest	ProjectFlip.UserInterface.Sur
Passed	ConvertBackTest	ProjectFlip.UserInterface.Sur
Passed	NotifyTest	ProjectFlip.UserInterface.Sur
Passed	OnShowFilterTest	ProjectFlip.UserInterface.Sur
Passed	OnCurrentMainCriteriaChangedTest	ProjectFlip.UserInterface.Sur
Passed	ExtractImageInvalidPathTest	ProjectFlip.Preparer.Test
Passed	TestRequirementsNotOkTestInvalidFromPath	ProjectFlip.Converter.Test
Passed	MatchTest	ProjectFlip.Services.Test
Passed	MoveToNextTest	ProjectFlip.UserInterface.Sur
Passed	ImportTest	ProjectFlip.Services.Loader.T
Passed	OnShowDetailTest	ProjectFlip.UserInterface.Sur
Passed	ConvertTest	ProjectFlip.Converter.Test
Passed	TestRequirementsNotOkTestInvalidToPath	ProjectFlip.Converter.Test
Passed	OverviewWindowViewModelProjectNotesTest	ProjectFlip.UserInterface.Sur
Passed	RaiseCanExecuteChangedTest	ProjectFlip.UserInterface.Sur
Passed	ExtractImageTest	ProjectFlip.Preparer.Test
Passed	ExecuteTest	ProjectFlip.UserInterface.Sur
Passed	TestRequirementsOk	ProjectFlip.Converter.Test
Passed	ProjectNotesServiceWithPnTest	ProjectFlip.Services.Test
Passed	AddAndRemoveAndCallbackFilterTest	ProjectFlip.UserInterface.Sur
Passed	ToListTests	ProjectFlip.Services.Test
Passed	MoveToPreviousTest	ProjectFlip.UserInterface.Sur
Passed	ConvertTest	ProjectFlip.UserInterface.Sur

Abbildung 62 - Unit Tests 15.11.2011

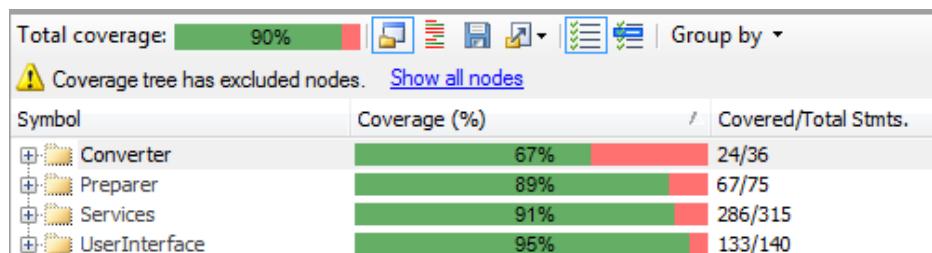
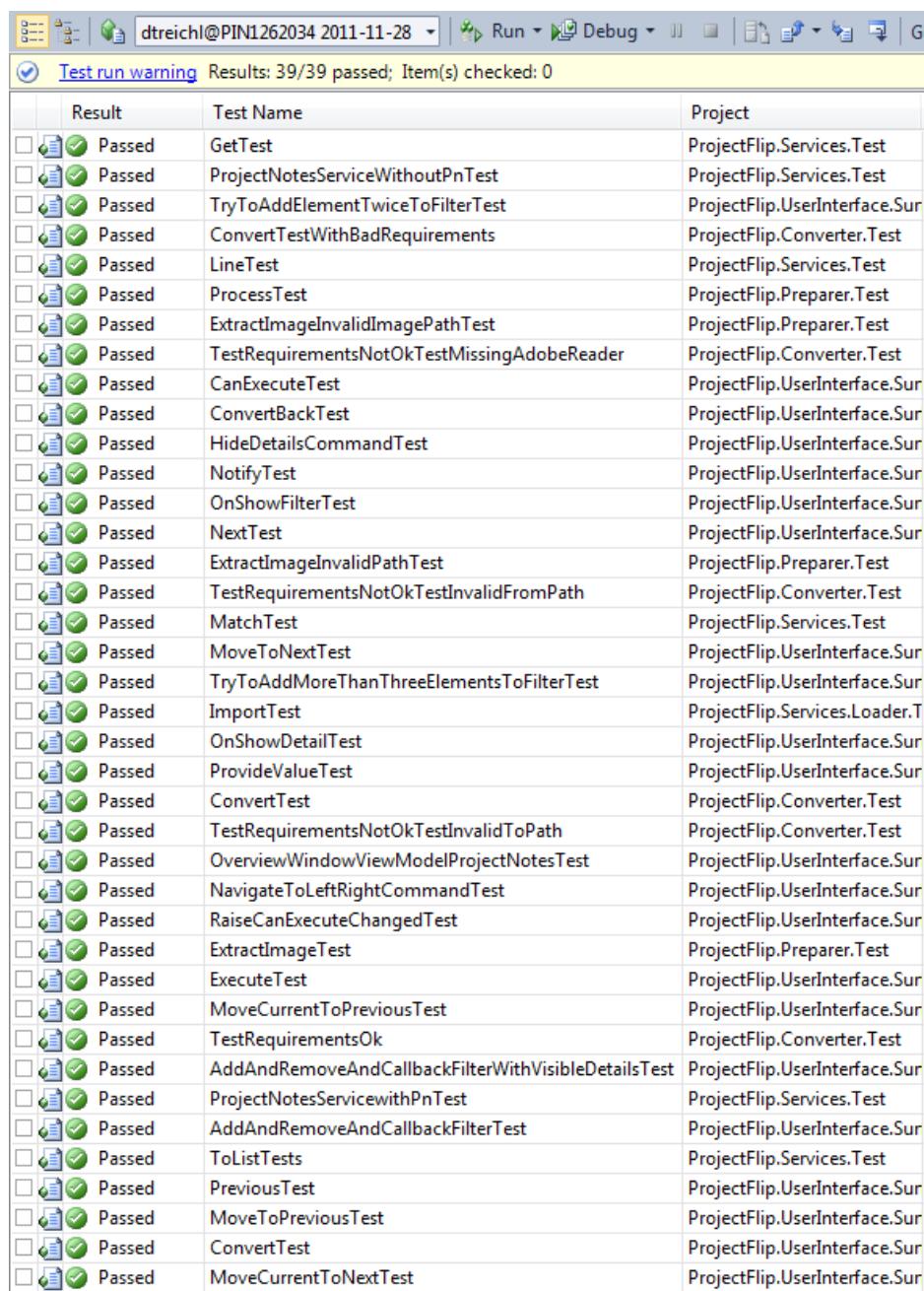


Abbildung 63 - Test Coverage 15.11.2011

V.8.2.1.3 Sprint 5



The screenshot shows a test results grid with three columns: Result, Test Name, and Project. The 'Result' column contains icons for Passed (green checkmark) and Failed (red X). The 'Test Name' column lists various test methods such as GetTest, ProjectNotesServiceWithoutPnTest, TryToAddElementTwiceToFilterTest, etc. The 'Project' column shows the corresponding project names for each test.

Result	Test Name	Project
Passed	GetTest	ProjectFlip.Services.Test
Passed	ProjectNotesServiceWithoutPnTest	ProjectFlip.Services.Test
Passed	TryToAddElementTwiceToFilterTest	ProjectFlip.UserInterface.Sur
Passed	ConvertTestWithBadRequirements	ProjectFlip.Converter.Test
Passed	LineTest	ProjectFlip.Services.Test
Passed	ProcessTest	ProjectFlip.Preparer.Test
Passed	ExtractImageInvalidImagePathTest	ProjectFlip.Preparer.Test
Passed	TestRequirementsNotOkTestMissingAdobeReader	ProjectFlip.Converter.Test
Passed	CanExecuteTest	ProjectFlip.UserInterface.Sur
Passed	ConvertBackTest	ProjectFlip.UserInterface.Sur
Passed	HideDetailsCommandTest	ProjectFlip.UserInterface.Sur
Passed	NotifyTest	ProjectFlip.UserInterface.Sur
Passed	OnShowFilterTest	ProjectFlip.UserInterface.Sur
Passed	NextTest	ProjectFlip.UserInterface.Sur
Passed	ExtractImageInvalidPathTest	ProjectFlip.Preparer.Test
Passed	TestRequirementsNotOkTestInvalidFromPath	ProjectFlip.Converter.Test
Passed	MatchTest	ProjectFlip.Services.Test
Passed	MoveToNextTest	ProjectFlip.UserInterface.Sur
Passed	TryToAddMoreThanThreeElementsToFilterTest	ProjectFlip.UserInterface.Sur
Passed	ImportTest	ProjectFlip.Services.Loader.T
Passed	OnShowDetailTest	ProjectFlip.UserInterface.Sur
Passed	ProvideValueTest	ProjectFlip.UserInterface.Sur
Passed	ConvertTest	ProjectFlip.Converter.Test
Passed	TestRequirementsNotOkTestInvalidToPath	ProjectFlip.Converter.Test
Passed	OverviewWindowViewModelProjectNotesTest	ProjectFlip.UserInterface.Sur
Passed	NavigateToLeftRightCommandTest	ProjectFlip.UserInterface.Sur
Passed	RaiseCanExecuteChangedTest	ProjectFlip.UserInterface.Sur
Passed	ExtractImageTest	ProjectFlip.Preparer.Test
Passed	ExecuteTest	ProjectFlip.UserInterface.Sur
Passed	MoveCurrentToPreviousTest	ProjectFlip.UserInterface.Sur
Passed	TestRequirementsOk	ProjectFlip.Converter.Test
Passed	AddAndRemoveAndCallbackFilterWithVisibleDetailsTest	ProjectFlip.UserInterface.Sur
Passed	ProjectNotesServiceWithPnTest	ProjectFlip.Services.Test
Passed	AddAndRemoveAndCallbackFilterTest	ProjectFlip.UserInterface.Sur
Passed	ToListTests	ProjectFlip.Services.Test
Passed	PreviousTest	ProjectFlip.UserInterface.Sur
Passed	MoveToPreviousTest	ProjectFlip.UserInterface.Sur
Passed	ConvertTest	ProjectFlip.UserInterface.Sur
Passed	MoveCurrentToNextTest	ProjectFlip.UserInterface.Sur

Abbildung 64 - Unit Tests 28.11.2011

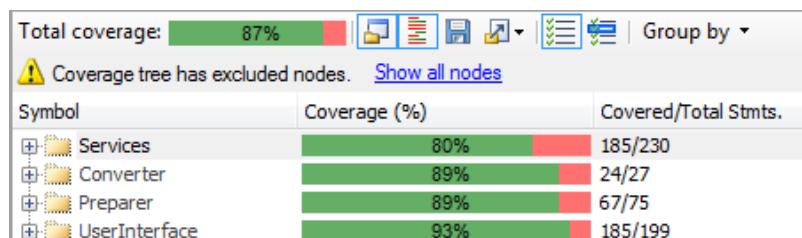


Abbildung 65 - Test Coverage 28.11.2011

V.8.2.1.4 Sprint 6

<input checked="" type="checkbox"/> Test run warning	Results: 47/47 passed; Item(s) checked: 0	
Result	Test Name	Project
<input type="checkbox"/>	Passed AddAndRemoveAndCallbackFilterTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed AddAndRemoveAndCallbackFilterWithVisibleDataTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed AggregatorNullTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed CanExecuteTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed CompareTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed ConvertBackTest	ProjectFlip.UserInterface.Surface.Converter.Test
<input type="checkbox"/>	Passed ConvertTest	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed ConvertTest	ProjectFlip.UserInterface.Surface.Converter.Test
<input type="checkbox"/>	Passed ConvertTestWithBadRequirements	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed ExecuteTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed ExtractImageInvalidImagePathTest	ProjectFlip.Preparer.Test
<input type="checkbox"/>	Passed ExtractImageWithPathTest	ProjectFlip.Preparer.Test
<input type="checkbox"/>	Passed ExtractImageTest	ProjectFlip.Preparer.Test
<input type="checkbox"/>	Passed GetTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed HideDetailsCommandTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed ImageLoadingTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed ImportTest	ProjectFlip.Services.Loader.Test
<input type="checkbox"/>	Passed LineTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed MatchTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed MoveCurrentToNextTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed MoveCurrentToPreviousTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed MoveToNextTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed MoveToPreviousTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed NavigateToLeftRightCommandTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed NextTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed NotifyTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed OnCurrentMainCriteriaChangedTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed OnShowDetailTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed OnShowFilterTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed OverviewWindowViewModelConstructorTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed OverviewWindowViewModelProjectNotesTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed PreviousTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed ProcessTest	ProjectFlip.Preparer.Test
<input type="checkbox"/>	Passed ProjectNotesServiceWithoutPnTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed ProjectNotesServiceWithPnTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed ProvideValueTest	ProjectFlip.UserInterface.Surface.Converter.Test
<input type="checkbox"/>	Passed RaiseCanExecuteChangedTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed TestRequirementsNotOkTestInvalidFromPath	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed TestRequirementsNotOkTestInvalidToPath	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed TestRequirementsNotOkTestMissingAdobeReader	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed TestRequirementsOk	ProjectFlip.Converter.Test
<input type="checkbox"/>	Passed ToListTests	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed TryToAddElementTwiceToFilterTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed TryToAddMoreThanThreeElementsToFilterTest	ProjectFlip.UserInterface.Surface.Test
<input type="checkbox"/>	Passed UrlTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed Xps2LoadingTest	ProjectFlip.Services.Test
<input type="checkbox"/>	Passed XpsLoadingTest	ProjectFlip.Services.Test

Abbildung 66 - Unit Tests 11.12.2011

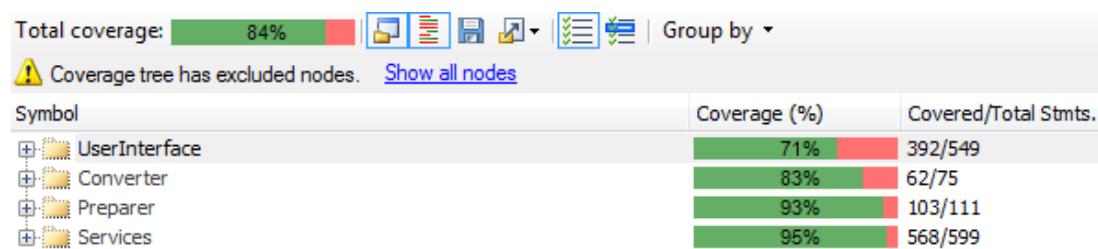


Abbildung 67 - Test Coverage 11.12.2011

V.8.2.1.5 Sprint 7

Test Results			
Result	Test Name	Project	Error Message
Passed	OverviewWindowViewModelConstructorTest	ProjectFlip.UserInterface.Surface.Test	
Passed	TryToAddElementTwiceToFilterTest	ProjectFlip.UserInterface.Surface.Test	
Passed	CanExecuteTest	ProjectFlip.UserInterface.Surface.Test	
Passed	HideDetailsCommandTest	ProjectFlip.UserInterface.Surface.Test	
Passed	NotifyTest	ProjectFlip.UserInterface.Surface.Test	
Passed	OnShowFilterTest	ProjectFlip.UserInterface.Surface.Test	
Passed	OnCurrentMainCriteriaChangedTest	ProjectFlip.UserInterface.Surface.Test	
Passed	MoveToNextTest	ProjectFlip.UserInterface.Surface.Test	
Passed	TryToAddMoreThanThreeElementsToFilterTest	ProjectFlip.UserInterface.Surface.Test	
Passed	OnShowDetailTest	ProjectFlip.UserInterface.Surface.Test	
Passed	OverviewWindowViewModelProjectNotesTest	ProjectFlip.UserInterface.Surface.Test	
Passed	NavigateToLeftRightCommandTest	ProjectFlip.UserInterface.Surface.Test	
Passed	RaiseCanExecuteChangedTest	ProjectFlip.UserInterface.Surface.Test	
Passed	GravatarsViewModelConstructorTest	ProjectFlip.UserInterface.Surface.Test	
Passed	ExecuteTest	ProjectFlip.UserInterface.Surface.Test	
Passed	AddAndRemoveAndCallbackFilterWithVisibleDetailsTest	ProjectFlip.UserInterface.Surface.Test	
Passed	AddAndRemoveAndCallbackFilterTest	ProjectFlip.UserInterface.Surface.Test	
Passed	MoveToPreviousTest	ProjectFlip.UserInterface.Surface.Test	
Passed	ConvertBackTest	ProjectFlip.UserInterface.Surface.Converter.Tes	
Passed	ProvideValueTest	ProjectFlip.UserInterface.Surface.Converter.Tes	
Passed	ConvertTest	ProjectFlip.UserInterface.Surface.Converter.Tes	
Passed	GetTest	ProjectFlip.Services.Test	
Passed	AggregatorNullTest	ProjectFlip.Services.Test	
Passed	ProjectNotesServiceWithoutPnTest	ProjectFlip.Services.Test	
Passed	LineTest	ProjectFlip.Services.Test	
Passed	Xps2LoadingTest	ProjectFlip.Services.Test	
Passed	PreviousTest	ProjectFlip.Services.Test	
Passed	ImageUrlTest	ProjectFlip.Services.Test	
Passed	MatchTest	ProjectFlip.Services.Test	
Passed	XpsLoadingTest	ProjectFlip.Services.Test	
Passed	CompareTest	ProjectFlip.Services.Test	
Passed	MoveCurrentToPreviousTest	ProjectFlip.Services.Test	
Passed	NextTest	ProjectFlip.Services.Test	
Passed	UrlTest	ProjectFlip.Services.Test	
Passed	ProjectNotesServicewithPnTest	ProjectFlip.Services.Test	
Passed	GravatarServiceConstructorTest	ProjectFlip.Services.Test	
Passed	ImageLoadingTest	ProjectFlip.Services.Test	
Passed	PersonConstructorTest	ProjectFlip.Services.Test	
Passed	MoveCurrentToNextTest	ProjectFlip.Services.Test	

Abbildung 68 - Unit Tests 16.12.2011 Teil 1

Result	Test Name	Project
Passed	MoveCurrentToNextTest	ProjectFlip.Services.Test
Passed	ToListTests	ProjectFlip.Services.Test
Passed	ImportTest	ProjectFlip.Services.Loader.Test
Passed	ProcessTest	ProjectFlip.Preparer.Test
Passed	ExtractImageInvalidImagePathTest	ProjectFlip.Preparer.Test
Passed	ExtractImageInvalidPathTest	ProjectFlip.Preparer.Test
Passed	ExtractImageTest	ProjectFlip.Preparer.Test
Passed	ConvertTestWithBadRequirements	ProjectFlip.Converter.Test
Passed	TestRequirementsNotOkTestMissingAdobeReader	ProjectFlip.Converter.Test
Passed	TestRequirementsNotOkTestInvalidFromPath	ProjectFlip.Converter.Test
Passed	ConvertTest	ProjectFlip.Converter.Test
Passed	TestRequirementsNotOkTestInvalidToPath	ProjectFlip.Converter.Test
Passed	TestRequirementsOk	ProjectFlip.Converter.Test

Abbildung 69 - Unit Tests 16.12.2011 Teil 2

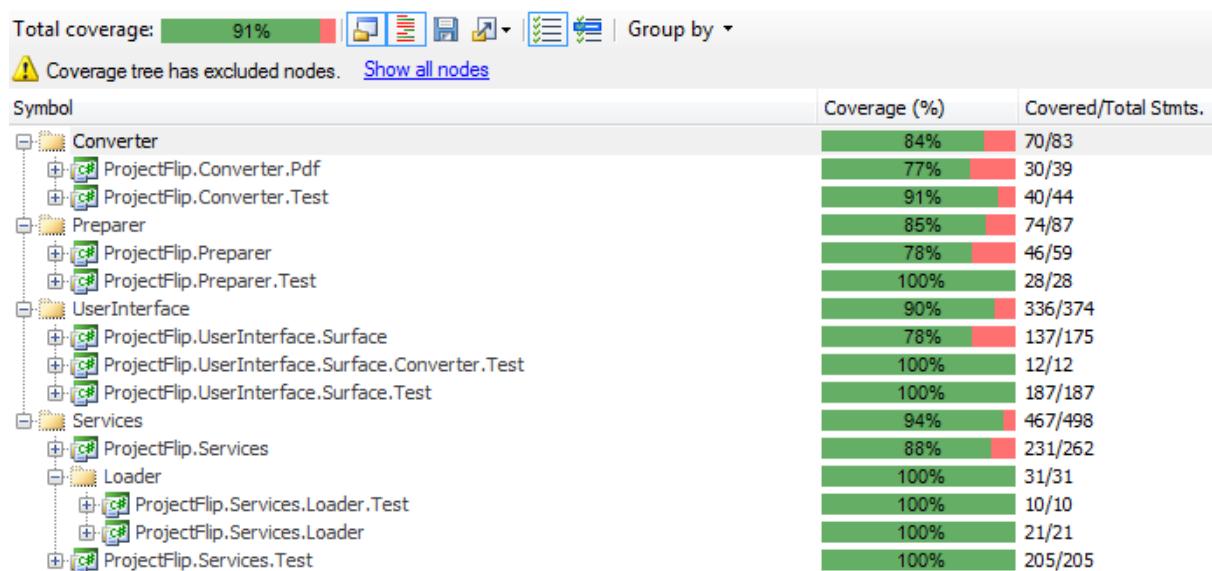


Abbildung 70 - Test Coverage 16.12.2011

V.8.2.2 System Tests

Die System Tests orientieren sich an den definierten User Stories, die dann im entsprechenden Sprint umgesetzt wurden.

V.8.2.2.1 Sprint 3

Testperson: Christina Heidt

#	Titel	Beschreibung	Resultat	Datum
1	Übersicht für PN	Die PNs lassen sich in der Übersicht anzeigen	Ok	28.10.2011
2	Detailansicht PN	Die Detailansicht einer PN lässt sich anzeigen und die PN kann gelesen werden	Ok	28.10.2011
3	Navigation „Übersicht -> Detail“	Der Nutzer kann von der Übersicht zur Einzelansicht wechseln	Ok	28.10.2011
4	Navigation „Detail“ -> „Übersicht“	Der Nutzer kann von der Einzelansicht in die Übersicht wechseln	Ok	28.10.2011
5	Darstellung der verkleinerten PN mit Bild	Der Nutzer sieht in der Übersicht zu jeder PN ein kleines Bild.	Ok	28.10.2011
6	Navigation "Detailansicht -> Detailansicht"	Der Nutzer kann von einer PN in der Detailansicht zur nächsten wechseln und sich diese ansehen.	Ok	28.10.2011

Tabelle 8 - System Tests Sprint 3

V.8.2.2.2 Sprint 4

Testperson: Christina Heidt

#	Titel	Beschreibung	Resultat	Datum
1	Übersicht für PN	Die PNs lassen sich in der Übersicht anzeigen	Ok	14.11.2011
2	Detailansicht PN	Die Detailansicht einer PN lässt sich anzeigen und die PN kann gelesen werden	Ok	14.11.2011
3	Navigation „Übersicht -> Detail“	Der Nutzer kann von der Übersicht zur Einzelansicht wechseln	Ok	14.11.2011
4	Navigation „Detail“ -> „Übersicht“	Der Nutzer kann von der Einzelansicht in die Übersicht wechseln	Ok	14.11.2011
5	Darstellung der verkleinerten PN mit Bild	Der Nutzer sieht in der Übersicht zu jeder PN ein kleines Bild.	Ok	14.11.2011
6	Navigation "Detailansicht -> Detailansicht"	Der Nutzer kann von einer PN in der Detailansicht zur nächsten wechseln und sich diese ansehen.	Ok	14.11.2011
7	Tags zu PN angezeigt	zu einer PN zusätzliche Begriffe anzeigen, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Ok	14.11.2011
8	Tags aggregiert	bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Ok	15.11.2011
9	Aggregierte Tags anpassbar	Als Surface Admin möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.	Ok	14.11.2011
10	Filter in Übersicht setzen	einen Filter in der Übersicht setzen, damit die angezeigten PN eingeschränkt werden.	Ok	14.11.2011
11	Filterkriterium auswählen	das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den	Ok	14.11.2011

		gesuchten Begriff einfach finden kann.		
12	Filter in Übersicht entfernen	einen Filter in der Übersicht entfernen, damit mir mehr PN angezeigt werden.	Ok	14.11.2011
13	Filter in Detailansicht entfernen	einen Filter in der Detailansicht entfernen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	14.11.2011

Tabelle 9 - System Tests Sprint 4

V.8.2.2.3 Sprint 5

Testperson: Delia Treichler

#	Titel	Beschreibung	Resultat	Datum
1	Übersicht für PN	Die PNs lassen sich in der Übersicht anzeigen	Ok	28.11.2011
2	Detailansicht PN	Die Detailansicht einer PN lässt sich anzeigen und die PN kann gelesen werden	Ok	28.11.2011
3	Navigation „Übersicht -> Detail“	Der Nutzer kann von der Übersicht zur Einzelansicht wechseln	Ok	28.11.2011
4	Navigation „Detail“ -> „Übersicht“	Der Nutzer kann von der Einzelansicht in die Übersicht wechseln	Ok	28.11.2011
5	Darstellung der verkleinerten PN mit Bild	Der Nutzer sieht in der Übersicht zu jeder PN ein kleines Bild.	Ok	28.11.2011
6	Navigation "Detailansicht -> Detailansicht"	Der Nutzer kann von einer PN in der Detailansicht zur nächsten wechseln und sich diese ansehen.	Ok	28.11.2011
7	Tags zu PN angezeigt	zu einer PN zusätzliche Begriffe anzeigen, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Ok	28.11.2011
8	Tags aggregiert	bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Ok	28.11.2011
9	Aggregierte Tags anpassbar	Als Surface Admin möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.	Ok	28.11.2011
10	Filter in Übersicht setzen	einen Filter in der Übersicht setzen, damit die angezeigten PN eingeschränkt werden.	Ok	28.11.2011
11	Filterkriterium auswählen	das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann.	Ok	28.11.2011
12	Filter in Übersicht entfernen	einen Filter in der Übersicht entfernen, damit mir mehr PN angezeigt werden.	Ok	28.11.2011
13	Filter in Detailansicht entfernen	einen Filter in der Detailansicht entfernen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	28.11.2011
14	Filter in Detailansicht setzen	einen Filter in der Detailansicht setzen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	28.11.2011
15	Filtern nach mehreren Kriterien	nach mehreren bestimmten Kriterien filtern, um eine begrenzte Auswahl zu erhalten.	Ok	28.11.2011
16	Animierte Navigation "Detail -> Übersicht"	in der Detailansicht erkennen können (z.B. durch Animation), damit ich den Weg von der Detailansicht zur Übersicht finde.	Ok	28.11.2011
17	Scrollerkennung in Übersicht dargestellt	an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann.	Ok	28.11.2011
18	Schöne Darstellung der	die zu einer PN zugehörigen Begriffe schön	Ok	28.11.2011

Tags	darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.		
19 Schöne Darstellung des Filters	den Filter schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Ok	28.11.2011
20 Animation für Navigation "Detailansicht -> Detailansicht"	eine Animation sehen, wenn ich durch die PN's navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat.	Ok	28.11.2011
21 Lesemode PN anzeigen	eine PN vergrößern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann.	Funktioniert noch nicht vollständig, timeboxed.	28.11.2011

Tabelle 10 - System Tests Sprint 5

V.8.2.2.4 Sprint 6

Testperson: Lukas Elmer

#	Titel	Beschreibung	Resultat	Datum
1	Übersicht für PN	Die PN's lassen sich in der Übersicht anzeigen	Ok	11.12.2011
2	Detailansicht PN	Die Detailansicht einer PN lässt sich anzeigen und die PN kann gelesen werden	Ok	11.12.2011
3	Navigation „Übersicht -> Detail“	Der Nutzer kann von der Übersicht zur Einzelansicht wechseln	Ok	11.12.2011
4	Navigation „Detail“ -> „Übersicht“	Der Nutzer kann von der Einzelansicht in die Übersicht wechseln	Ok	11.12.2011
5	Darstellung der verkleinerten PN mit Bild	Der Nutzer sieht in der Übersicht zu jeder PN ein kleines Bild.	Ok	11.12.2011
6	Navigation "Detailansicht -> Detailansicht"	Der Nutzer kann von einer PN in der Detailansicht zur nächsten wechseln und sich diese ansehen.	Ok	11.12.2011
7	Tags zu PN angezeigt	zu einer PN zusätzliche Begriffe anzeigen, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Ok	11.12.2011
8	Tags aggregiert	bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Ok	11.12.2011
9	Aggregierte Tags anpassbar	Als Surface Admin möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.	Ok	11.12.2011
10	Filter in Übersicht setzen	einen Filter in der Übersicht setzen, damit die angezeigten PN eingeschränkt werden.	Ok	11.12.2011
11	Filterkriterium auswählen	das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann.	Ok	11.12.2011
12	Filter in Übersicht entfernen	einen Filter in der Übersicht entfernen, damit mir mehr PN angezeigt werden.	Ok	11.12.2011
13	Filter in Detailansicht entfernen	einen Filter in der Detailansicht entfernen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	11.12.2011
14	Filter in Detailansicht setzen	einen Filter in der Detailansicht setzen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	11.12.2011
15	Filtern nach mehreren	nach mehreren bestimmten Kriterien filtern, um	Ok	11.12.2011

Kriterien		eine begrenzte Auswahl zu erhalten.		
16	Animierte Navigation "Detail -> Übersicht"	in der Detailansicht erkennen können (z.B. durch Animation), damit ich den Weg von der Detailansicht zur Übersicht finde.	Ok	11.12.2011
17	Scrollerkennung in Übersicht dargestellt	an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann.	Ok	11.12.2011
18	Schöne Darstellung der Tags	die zu einer PN zugehörigen Begriffe schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Ok	11.12.2011
19	Schöne Darstellung des Filters	den Filter schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Ok	11.12.2011
20	Animation für Navigation "Detailansicht -> Detailansicht"	eine Animation sehen, wenn ich durch die PNs navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat.	Ok	11.12.2011
21	Lesemodus PN anzeigen	eine PN vergrössern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann.	Ok	11.12.2011
22	Easteregg / Info View	die Info View öffnen und schliessen, damit ich sehen kann, wie die Applikation entstanden ist und wer sie implementiert hat.	Ok	11.12.2011
23	Anzahl PN bei Übersicht angezeigt	die Anzahl PN in der Übersicht sehen, damit ich weiss, wie gross die Liste ist.	Ok	11.12.2011
24	Animation Filter -> Übersicht	eine Animation sehen, wenn die Ansicht geschlossen und der Filter gesetzt wird, damit ich sehe, dass und wo der Filter nun gesetzt ist. (Filter hervorheben, nach dem setzen des Kriteriums, gemäss Protokoll)	Ok	11.12.2011

Tabelle 11 - System Tests Sprint 6

V.8.2.2.5 Sprint 7

Testperson: Lukas Elmer

#	Titel	Beschreibung	Resultat	Datum
1	Übersicht für PN	Die PNs lassen sich in der Übersicht anzeigen	Ok	16.12.2011
2	Detailansicht PN	Die Detailansicht einer PN lässt sich anzeigen und die PN kann gelesen werden	Ok	16.12.2011
3	Navigation „Übersicht -> Detail“	Der Nutzer kann von der Übersicht zur Einzelansicht wechseln	Ok	16.12.2011
4	Navigation „Detail“ -> „Übersicht“	Der Nutzer kann von der Einzelansicht in die Übersicht wechseln	Ok	16.12.2011
5	Darstellung der verkleinerten PN mit Bild	Der Nutzer sieht in der Übersicht zu jeder PN ein kleines Bild.	Ok	16.12.2011
6	Navigation "Detailansicht -> Detailansicht"	Der Nutzer kann von einer PN in der Detailansicht zur nächsten wechseln und sich diese ansehen.	Ok	16.12.2011
7	Tags zu PN angezeigt	zu einer PN zusätzliche Begriffe anzeigen, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Ok	16.12.2011
8	Tags aggregiert	bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Ok	16.12.2011
9	Aggregierte Tags	Als Surface Admin möchte ich die Zuordnungen	Ok	16.12.2011

	anpassbar	der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.		
10	Filter in Übersicht setzen	einen Filter in der Übersicht setzen, damit die angezeigten PN eingeschränkt werden.	Ok	16.12.2011
11	Filterkriterium auswählen	das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann.	Ok	16.12.2011
12	Filter in Übersicht entfernen	einen Filter in der Übersicht entfernen, damit mir mehr PN angezeigt werden.	Ok	16.12.2011
13	Filter in Detailansicht entfernen	einen Filter in der Detailansicht entfernen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	16.12.2011
14	Filter in Detailansicht setzen	einen Filter in der Detailansicht setzen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	Ok	16.12.2011
15	Filtern nach mehreren Kriterien	nach mehreren bestimmten Kriterien filtern, um eine begrenzte Auswahl zu erhalten.	Ok	16.12.2011
16	Animierte Navigation "Detail -> Übersicht"	in der Detailansicht erkennen können (z.B. durch Animation), damit ich den Weg von der Detailansicht zur Übersicht finde.	Ok	16.12.2011
17	Scrollerkennung in Übersicht dargestellt	an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann.	Ok	16.12.2011
18	Schöne Darstellung der Tags	die zu einer PN zugehörigen Begriffe schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Ok	16.12.2011
19	Schöne Darstellung des Filters	den Filter schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Ok	16.12.2011
20	Animation für Navigation "Detailansicht -> Detailansicht"	eine Animation sehen, wenn ich durch die PNs navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat.	Ok	16.12.2011
21	Lesemodus PN anzeigen	eine PN vergrössern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann.	Ok	16.12.2011
22	Easteregg / Info View	die Info View öffnen und schliessen, damit ich sehen kann, wie die Applikation entstanden ist und wer sie implementiert hat.	Ok	16.12.2011
23	Anzahl PN bei Übersicht angezeigt	die Anzahl PN in der Übersicht sehen, damit ich weiss, wie gross die Liste ist.	Ok	16.12.2011
24	Animation Filter -> Übersicht	eine Animation sehen, wenn die Ansicht geschlossen und der Filter gesetzt wird, damit ich sehe, dass und wo der Filter nun gesetzt ist. (Filter hervorheben, nach dem setzen des Kriteriums, gemäss Protokoll)	Ok	16.12.2011

Tabelle 12 - System Tests Sprint 7

V.8.2.3 Usability Test

Mit der Durchführung eines summativen Tests wurde die Usability der Applikation geprüft. Die gewonnenen Erkenntnisse wurden protokolliert und in der Applikation entsprechend angepasst.

Die ausgefüllten Original-Fragebögen sind im Anhang C zu finden.

V.8.2.3.1 Test Setup

Die Anwendung wird mit einem HP LD4200tm Touch-Monitor getestet, da der Surface 2 nicht verfügbar ist. Der Monitor wird stehend platziert. Die Testpersonen müssen einzeln den Raum betreten und erhalten dort die Aufgabenstellung. Diese soll maximal fünf Minuten studiert werden. Die Anleitung muss im Verlauf des Tests nicht abgegeben werden, da sich dieser in 3 Unteraufgaben gliedert.

V.8.2.3.2 Testszenario

Sie sind als Mitarbeiter der Firma Swisscom spezialisiert auf Projekte im Bereich .NET. Heute haben Sie einen Termin bei der Zühlke Engineering AG, welche ihr Projektpartner ist. Es ist das erste Meeting, daher treffen Sie etwas verfrüh ein. Die Empfangsdame weist Sie darauf hin, dass die Project Notes der Firma (Projektbeschrieb auf eine A4-Seite zusammengefasst) auf dem neu erworbenen Surface eingesehen werden können.

V.8.2.3.2.1 Aufgabe 1

Sie nutzen daher die Wartezeit, um sich mit der neuen Technologie zu befassen. Gleichzeitig sind Sie neugierig herauszufinden, ob die Zühlke Engineering AG Erfahrungen in Ihrem Spezialgebiet .NET mitbringt.

V.8.2.3.2.2 Aufgabe 2

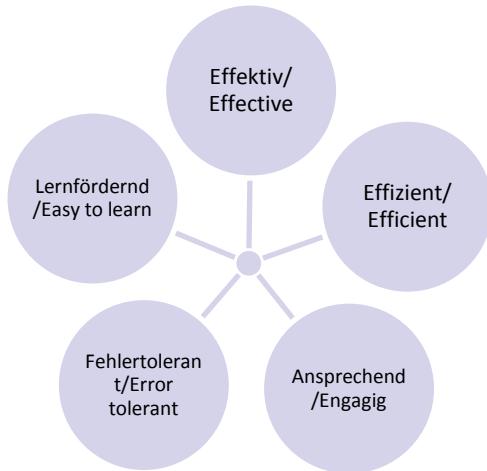
Einer Ihrer Arbeitskollegen der Swisscom erzählte Ihnen, dass er ein Projekt mit der Zühlke Engineering AG gemacht hatte. Sie mögen sich noch entfernt erinnern, um was es ungefähr ging, Ihnen fällt aber kein konkreter Begriff zu diesem Projekt ein. Trotzdem möchten Sie nun mehr über dieses Projekt herausfinden.

V.8.2.3.2.3 Aufgabe 3

Sie möchten sich nun nochmals einen Überblick über alle Projekte verschaffen.

V.8.2.3.3 Kriterien

Die Applikation wurde auf die Benutzbarkeit getestet, nach den Kriterien der Norm ISO 9241-11 und Whitney Quesenberry.



- **Effektivität:** Genauigkeit und Vollständigkeit mit der Benutzer ein bestimmtes Ziel erreichen
- **Effizienz:** Aufwand der Benutzer im Verhältnis zur Genauigkeit und Vollständigkeit des erzielten Effekts
- **Zufriedenheit:** positive Einstellung der Benutzer gegenüber der Nutzung des Systems. Keine Beeinträchtigungen der Freiheit durch das System. Dieser Punkt beinhaltet:
 - Lernfördernd
 - Fehlertolerant
 - Ansprechend

Abbildung 71 - Usability nach ISO 9241-11 und Quesenberry

V.8.2.3.4 Testpersonen

Testperson	Durchführungstermin	Status
Michael Gfeller	05.12.2011	durchgeführt
Mirco Stocker	05.12.2011	durchgeführt
Markus Stolze	05.12.2011	durchgeführt
Mischa Trecco	05.12.2011	durchgeführt

Tabelle 13 - Durchführungen Usability Tests

Die Fragebogen wurden von den Testpersonen ausgefüllt und wieder in dieses Dokument eingetragen. Kopien des Originalfragebogens befinden sich im Anhang (siehe VII Anhang).

V.8.2.3.5 Fragebogen Michael Gfeller

Bezeichnung		++	+	=	-	--	kinb
++	Trifft in hohem Mass zu	-		Trifft in geringem Mass zu			
+	Trifft grösstenteils zu	--		Trifft gar nicht zu			
=	Trifft mehr oder weniger zu	kinb		Kann ich nicht beurteilen			

A Effektivität		++	+	=	-	--	kinb
A1	Der Themenbereich kann eingeschränkt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A2	Gewünschte Project Notes können gefunden werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A3	Es können wieder alle Projekte angezeigt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A4	Es kann durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A5	Es ist ersichtlich wie viele Projekte gerade sichtbar sind	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A6	Eine Project Note kann gelesen werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A7	Eine Project Note kann vergrössert und besser gelesen werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

B Effizienz		++	+	=	-	--	kinb
B1	Der Themenbereich kann einfach eingeschränkt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B2	Project Notes eines bestimmten Themenbereichs können einfach gefunden werden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B3	Es können wieder einfach alle Projekte angezeigt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B4	Es kann einfach durch die Project Notes navigiert werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

C Fehlertoleranz		++	+	=	-	--	kinb
C1	Wird ein falsches Filterkriterium gesetzt, kann dieses wieder entfernt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C2	Das Programm stürzt nicht ab	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
C3	Beliebiges und zufälliges schnelles Klicken versetzt das Programm in keinen fehlerhaften Zustand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

D Zufriedenheit		++	+	=	-	--	kinb
D1	Die Applikation ist einfach zu benutzen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D2	Die Applikation ist angenehm zu bedienen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D3	Die einzelnen Felder sind selbsterklärend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

E Erlernbarkeit		++	+	=	-	--	kinb
E1	Die Applikation ist einfach zu bedienen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E2	Es ist schnell verständlich, wie die Applikation bedient werden muss	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E3	Die Applikation ist auch ohne Erklärung oder ein Handbuch verständlich	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

F Design		++	+	=	-	--	kinb
F1	Die graphische Oberfläche ist ansprechend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F2	Die graphische Oberfläche ist benutzerfreundlich	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F3	Texte sind gut lesbar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F4	Die einzelnen Schaltflächen sind einfach zu treffen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentare und Verbesserungsvorschläge

Ev. Filtern schneller, z.B. Virtualisieren der Panels.

Löschen der Kriterien via „wegwischen“

V.8.2.3.6 Fragebogen Mirko Stocker

Bezeichnung							
++	Trifft in hohem Mass zu	-	Trifft in geringem Mass zu				
+	Trifft grösstenteils zu	--	Trifft gar nicht zu				
=	Trifft mehr oder weniger zu	kinb	Kann ich nicht beurteilen				

A Effektivität		++	+	=	-	--	kinb
A1	Der Themenbereich kann eingeschränkt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A2	Gewünschte Project Notes können gefunden werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A3	Es können wieder alle Projekte angezeigt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A4	Es kann durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A5	Es ist ersichtlich wie viele Projekte gerade sichtbar sind	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A6	Eine Project Note kann gelesen werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A7	Eine Project Note kann vergrössert und besser gelesen werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

B Effizienz		++	+	=	-	--	kinb
B1	Der Themenbereich kann einfach eingeschränkt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B2	Project Notes eines bestimmten Themenbereichs können einfach gefunden werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B3	Es können wieder einfach alle Projekte angezeigt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B4	Es kann einfach durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

C Fehlertoleranz		++	+	=	-	--	kinb
C1	Wird ein falsches Filterkriterium gesetzt, kann dieses wieder entfernt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C2	Das Programm stürzt nicht ab	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C3	Beliebiges und zufälliges schnelles Klicken versetzt das Programm in keinen fehlerhaften Zustand	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

D Zufriedenheit		++	+	=	-	--	kinb
D1	Die Applikation ist einfach zu benutzen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D2	Die Applikation ist angenehm zu bedienen	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D3	Die einzelnen Felder sind selbsterklärend	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

E Erlernbarkeit		++	+	=	-	--	kinb
E1	Die Applikation ist einfach zu bedienen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E2	Es ist schnell verständlich, wie die Applikation bedient werden muss	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E3	Die Applikation ist auch ohne Erklärung oder ein Handbuch verständlich	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

F Design		++	+	=	-	--	kinb
F1	Die graphische Oberfläche ist ansprechend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F2	Die graphische Oberfläche ist benutzerfreundlich	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F3	Texte sind gut lesbar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F4	Die einzelnen Schaltflächen sind einfach zu treffen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentare und Verbesserungsvorschläge

V.8.2.3.7 Fragebogen Markus Stolze

Bezeichnung								
++	Trifft in hohem Mass zu	-	Trifft in geringem Mass zu					
+	Trifft grösstenteils zu	--	Trifft gar nicht zu					
=	Trifft mehr oder weniger zu	kinb	Kann ich nicht beurteilen					

A Effektivität		++	+	=	-	--	kinb
A1	Der Themenbereich kann eingeschränkt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A2	Gewünschte Project Notes können gefunden werden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A3	Es können wieder alle Projekte angezeigt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A4	Es kann durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A5	Es ist ersichtlich wie viele Projekte gerade sichtbar sind	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A6	Eine Project Note kann gelesen werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A7	Eine Project Note kann vergrössert und besser gelesen werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

B Effizienz		++	+	=	-	--	kinb
B1	Der Themenbereich kann einfach eingeschränkt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B2	Project Notes eines bestimmten Themenbereichs können einfach gefunden werden	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B3	Es können wieder einfach alle Projekte angezeigt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B4	Es kann einfach durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

C Fehlertoleranz		++	+	=	-	--	kinb
C1	Wird ein falsches Filterkriterium gesetzt, kann dieses wieder entfernt werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
C2	Das Programm stürzt nicht ab	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C3	Beliebiges und zufälliges schnelles Klicken versetzt das Programm in keinen fehlerhaften Zustand	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

D Zufriedenheit		++	+	=	-	--	kinb
D1	Die Applikation ist einfach zu benutzen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D2	Die Applikation ist angenehm zu bedienen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D3	Die einzelnen Felder sind selbsterklärend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

E Erlernbarkeit		++	+	=	-	--	kinb
E1	Die Applikation ist einfach zu bedienen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E2	Es ist schnell verständlich, wie die Applikation bedient werden muss	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E3	Die Applikation ist auch ohne Erklärung oder ein Handbuch verständlich	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

F Design		++	+	=	-	--	kinb
F1	Die graphische Oberfläche ist ansprechend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F2	Die graphische Oberfläche ist benutzerfreundlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
F3	Texte sind gut lesbar	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F4	Die einzelnen Schaltflächen sind einfach zu treffen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentare und Verbesserungsvorschläge	
Volltextsuche fehlt	
Die Applikation ist etwas langsam	

V.8.2.3.8 Fragebogen Mischa Trecco

Bezeichnung		++	+	=	-	--	kinb
++	Trifft in hohem Mass zu	-		Trifft in geringem Mass zu			
+	Trifft grösstenteils zu	--		Trifft gar nicht zu			
=	Trifft mehr oder weniger zu	kinb		Kann ich nicht beurteilen			

A Effektivität	++	+	=	-	--	kinb
A1 Der Themenbereich kann eingeschränkt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
A2 Gewünschte Project Notes können gefunden werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
A3 Es können wieder alle Projekte angezeigt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
A4 Es kann durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
A5 Es ist ersichtlich wie viele Projekte gerade sichtbar sind	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A6 Eine Project Note kann gelesen werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
A7 Eine Project Note kann vergrössert und besser gelesen werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

B Effizienz	++	+	=	-	--	kinb
B1 Der Themenbereich kann einfach eingeschränkt werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B2 Project Notes eines bestimmten Themenbereichs können einfach gefunden werden	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B3 Es können wieder einfach alle Projekte angezeigt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B4 Es kann einfach durch die Project Notes navigiert werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

C Fehlertoleranz	++	+	=	-	--	kinb
C1 Wird ein falsches Filterkriterium gesetzt, kann dieses wieder entfernt werden	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
C2 Das Programm stürzt nicht ab	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
C3 Beliebiges und zufälliges schnelles Klicken versetzt das Programm in keinen fehlerhaften Zustand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

D Zufriedenheit	++	+	=	-	--	kinb
D1 Die Applikation ist einfach zu benutzen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D2 Die Applikation ist angenehm zu bedienen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D3 Die einzelnen Felder sind selbsterklärend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

E Erlernbarkeit	++	+	=	-	--	kinb
E1 Die Applikation ist einfach zu bedienen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E2 Es ist schnell verständlich, wie die Applikation bedient werden muss	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E3 Die Applikation ist auch ohne Erklärung oder ein Handbuch verständlich	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

F Design	++	+	=	-	--	kinb
F1 Die graphische Oberfläche ist ansprechend	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F2 Die graphische Oberfläche ist benutzerfreundlich	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F3 Texte sind gut lesbar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F4 Die einzelnen Schaltflächen sind einfach zu treffen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentare und Verbesserungsvorschläge

- Sehr viele Kriterien. Es war unklar/nicht ersichtlich, dass ich scrollen kann.
Etwas zu viel Scrolling. Kategorien/Gruppen wäre evtl. hilfreich
- Wechsel zum Zoom-Modus eines Projektes komisch/unerwartet. Würde nicht automatisch zoomen, dafür Zoom-Button anzeigen.

V.8.2.3.9 Gesamtauswertung

Anhand der Antworten aus den Fragebögen konnten nachfolgende Auswertungen erstellt werden. Dabei wurden Fragen, die mit „kann ich nicht beurteilen“ angekreuzt wurden, ignoriert. Die Summe der einzelnen Fragen ergibt die Zufriedenheit zum jeweiligen Punkt in Prozent.

Die mit Buchstaben und Zahlen bezeichneten Legendeneinträge der Abbildungen beziehen sich auf die in den Fragebögen gestellten Fragen.

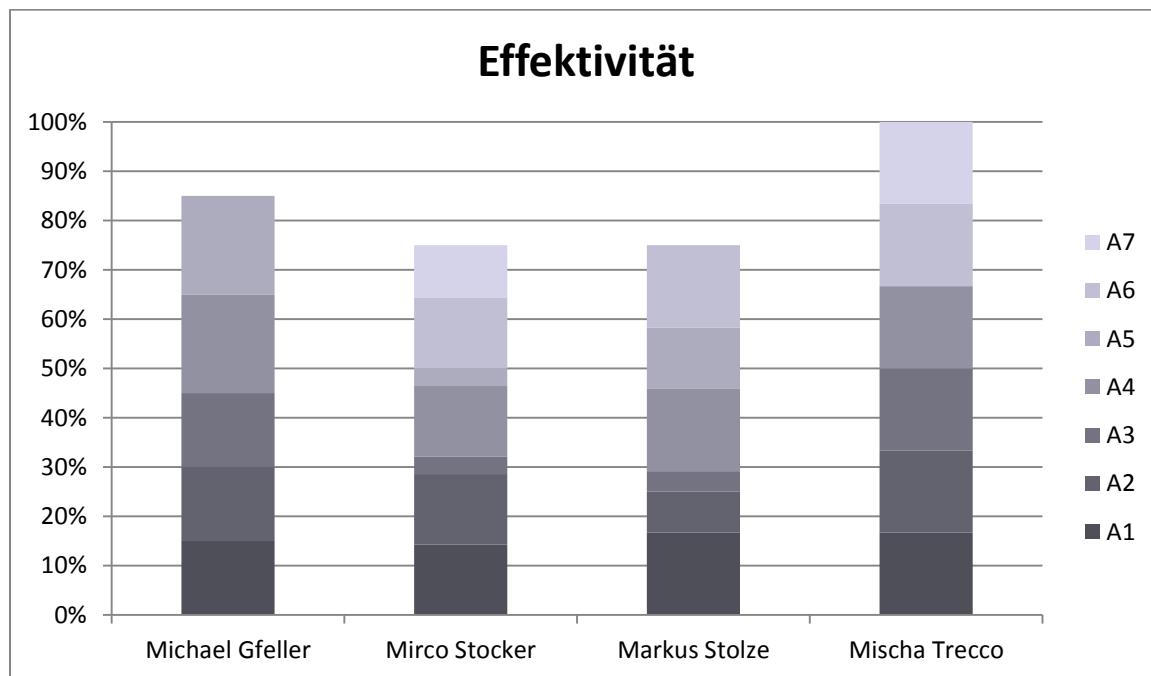


Abbildung 72 - Effektivität

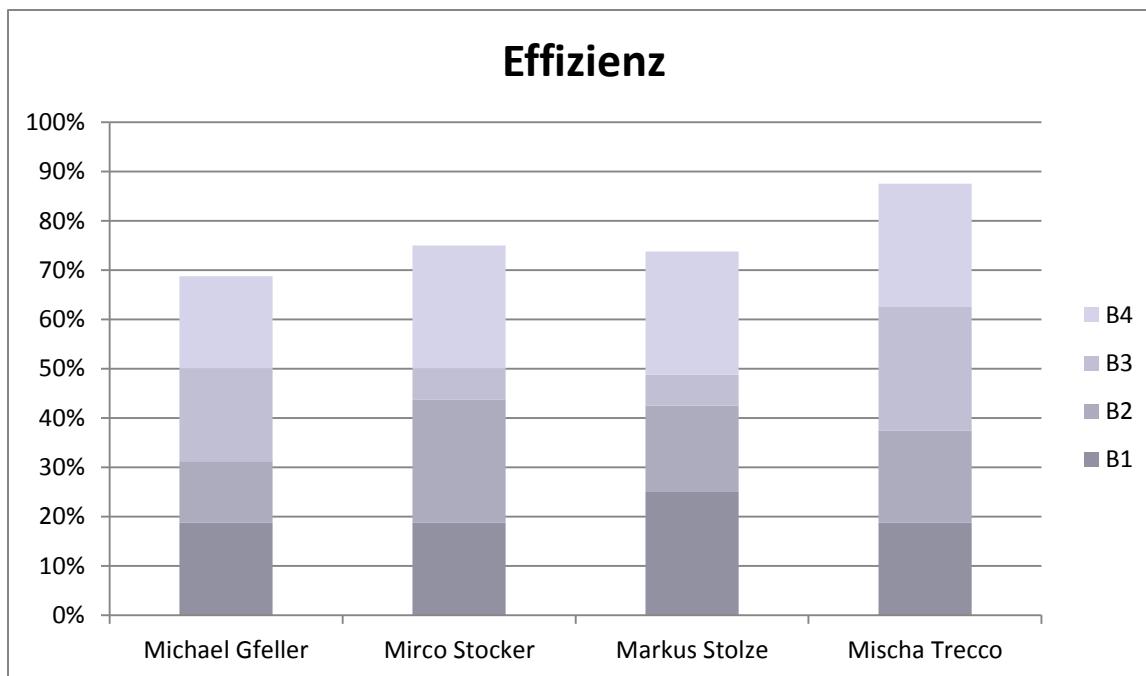


Abbildung 73 - Effizienz

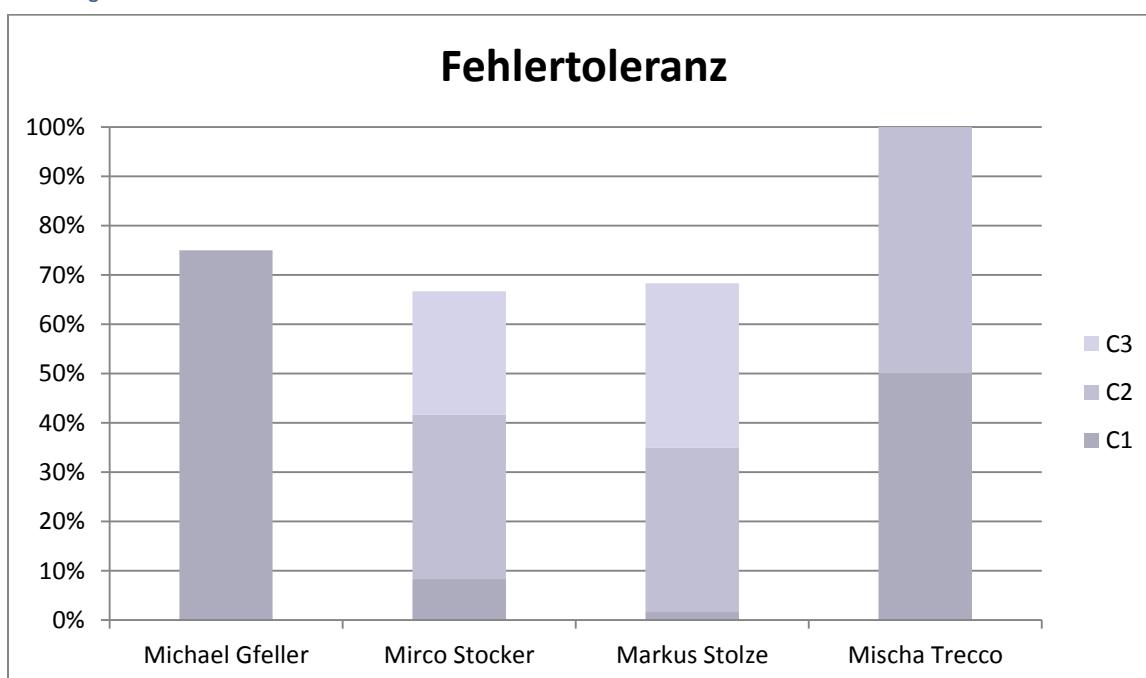


Abbildung 74 - Fehlertoleranz

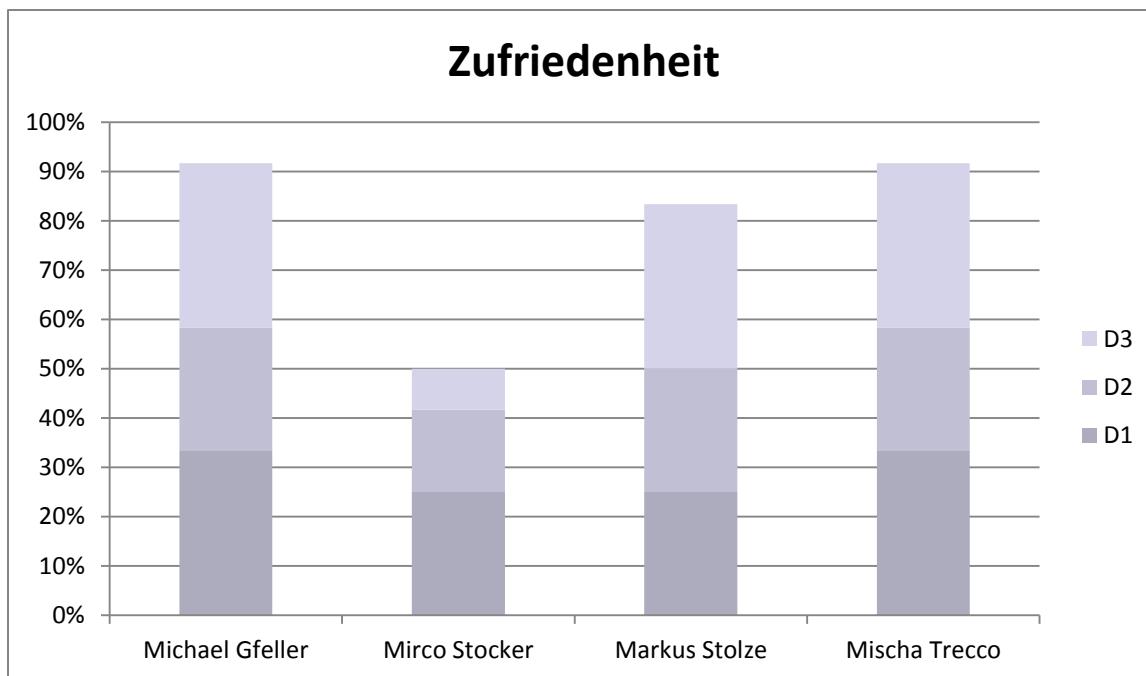


Abbildung 75 - Zufriedenheit

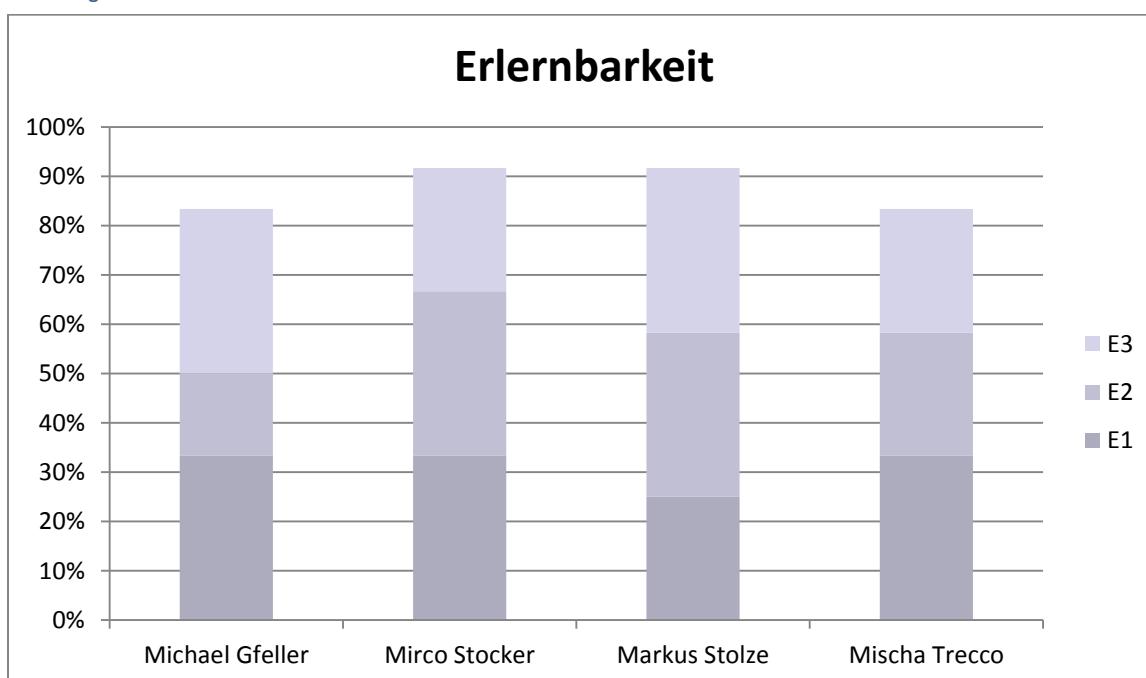


Abbildung 76 - Erlernbarkeit

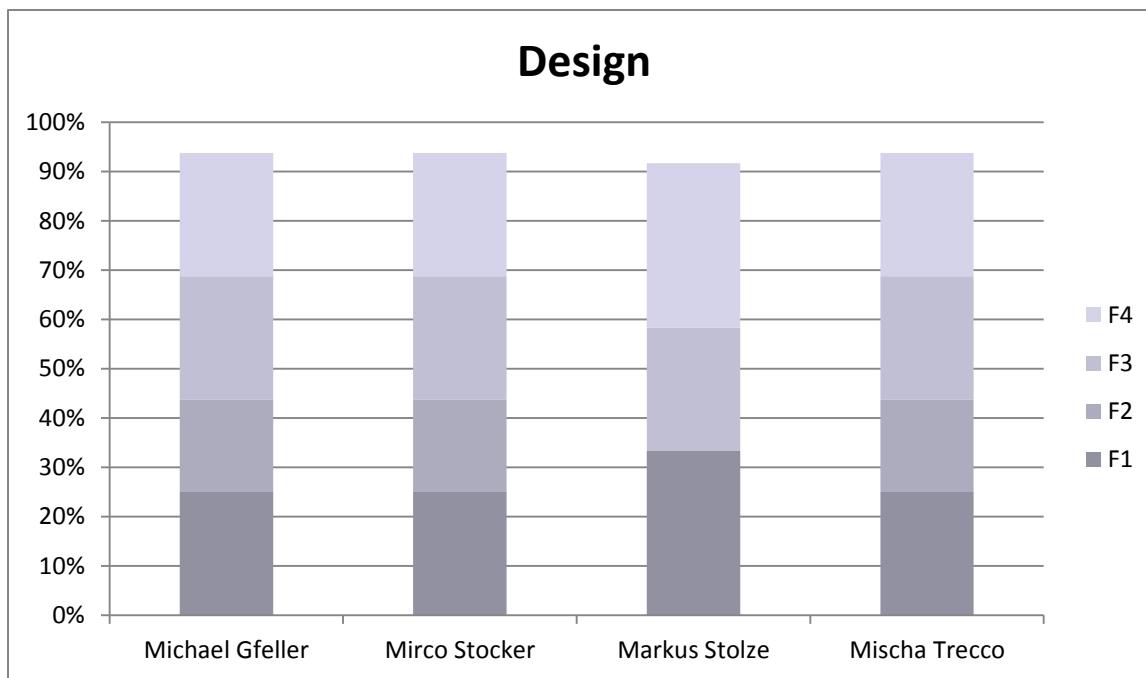


Abbildung 77 - Design

V.8.2.3.10 Fazit

Neben den Angaben aus den Fragebögen fliesst auch die Benutzerbeobachtung und Fehlerentdeckung während den Tests in das Fazit mit ein. Folgende Punkte wurden dabei erkannt und in der Applikation überarbeitet:

1. Bei über 300 Project Notes in der Übersicht konnte man sich plötzlich über die untere und obere Grenze des Scrollbereichs bewegen. Dies wurde vermutlich durch die Verwendung von Schlagschatten bei den Project Notes verursacht und konnte behoben werden.
2. Im Kriterien -Auswahlfenster war nicht ersichtlich, dass man scrollen kann. Deshalb wurde die Fenstergrösse so angepasst, dass nun ein Teil von den untersten, noch ersichtlichen Elementen vom Fensterrand überdeckt und somit abgeschnitten wird.
3. Die meisten Testpersonen erwarteten bei der Touch Applikation die Möglichkeit, Gesten benutzen zu können. Aus diesem Grund versuchten sie, die gesetzten Filterkriterien mit einem „swipe“ aus dem Filter zu entfernen. Die Gesten konnten jedoch durch den begrenzten Zeitrahmen nicht umgesetzt werden. Dadurch fielen die Bewertungen der Punkte A3 und B3 sowie C1 eher niedrig aus. Um klarer ersichtlich zu machen, wie das Filterkriterium entfernt werden kann, erscheint neu auf den ausgewählten Filterkriterien eine zusätzliche Schaltfläche.
4. Das Gleiche Problem mit den Gesten gilt für die Vergrösserung einer Project Note. Durch einen „single tab“ konnte die Project Note vergrössert und auch wieder verkleinert werden. Dieses Verhalten war für die Nutzer unerwartet und irritierte diese. Daher wurden Zoom-Buttons zum Vergrössern und Verkleinern der Project Note in die Applikation eingefügt. Auf der Project Note selbst kann nun nur noch gescrollt werden.
5. In der Übersicht über die Project Notes wurde nach dem Setzen oder Entfernen eines Filterkriteriums die Position der Scrollbar nicht zurückgesetzt. Wurde beispielsweise zuerst in der Übersicht ganz an das Ende der Liste gescrollt und dann die Projektauswahl eingeschränkt, so war die Position der Scrollbar nach der Einschränkung noch immer am Ende der Liste. Auch die Liste der Projektdaten und die Liste der Filterkriterien verhielten sich auf diese Weise. Dieses Problem wurde behoben.
6. Einige der Test-Benutzer übersahen, dass die Anzahl der Project Notes in der Applikation angezeigt wird. Um den Nutzer auf dieses Element aufmerksam zu machen, wird nun, wenn sich die Anzahl der Project Notes verändert, die aktuelle Zahl für kurze Zeit vergrössert angezeigt.
7. Mirko Stocker merkte zudem an, dass es praktisch wäre, wenn sich das Kriterien -Auswahlfenster auch öffnen würde, wenn man einen leeren Filter-Slot antippen würde. Dies wurde als sehr nützlich betrachtet und daher ebenfalls umgesetzt.

V.8.3 Betriebs- & Installationsdokumentation

V.8.3.1 Betriebskonzept

Da der Surface 2 zum Zeitpunkt dieses Projektes noch nicht verfügbar ist, beschränkt sich das Betriebskonzept auf Informationen, die schon jetzt bekannt sind. Das Betriebskonzept muss in Zukunft noch weiter ausgearbeitet werden.

V.8.3.1.1 Reinigung

Der Surface Touchtisch soll periodisch (minimal wöchentlich) gereinigt werden, um die Oberflächenverschmutzung infolge der Berührungen zu entfernen.

V.8.3.1.2 Aktualisierung der Projektreferenzen

Da immer wieder neue Projekte durch die Zühlke Engineering AG realisiert werden, sollten die Project Notes wöchentlich oder monatlich aktualisiert werden. Dies ist am besten über einen Cronjob zu realisieren. Dazu sollte dann auch eine Sharepoint Anbindung implementiert werden (siehe V.9.3 Ausblick).

V.8.3.1.3 Neustarten des Surface

Für die aktuelle Version der Applikation kann nicht garantiert werden, dass keine Memory Leaks vorhanden sind. Aus diesem Grund soll die Applikation täglich beendet und neu gestartet werden.

V.8.3.1.4 Mapping

Werden neue Project Notes hinzugefügt, muss die Aggregation der Kategorien und Begriffe überprüft werden. Das Mapping befindet sich in einem mit Tabulatoren getrennten Textfile, welches mithilfe von Microsoft Excel bearbeitet werden kann. Wo die Datei abgelegt werden wird und ob es andere Möglichkeiten zur Administration gibt, müssen bei der Installation und der Weiterentwicklung noch genau festgelegt werden.

V.8.3.2 Installationsanleitung

Da der Surface 2 zum Zeitpunkt dieses Projektes noch nicht verfügbar ist, beschränkt sich die Installationsanleitung auf Informationen, die schon jetzt bekannt sind. Für die tatsächliche Einführung muss diese Anleitung mit der Hardware getestet und eventuell angepasst werden.

V.8.3.2.1 Vorbedingungen

Es wird benötigt:

- Kompilierte Version des Projektes
- .NET, Surface 2 Runtime
- Adobe Reader X (v10)
- Internetverbindung

Der Pfad zum Adobe Reader muss über das Config File
ProjectFlip.Preparer/bin/Release/ProjectFlip.Preparer.exe.config eingestellt werden.

V.8.3.2.2 Installation

Um die Applikation zu installieren, müssen die kompilierten Dateien (exe, dll) in der aktuell kompilierten Ordnerhierarchie auf das Zielgerät kopiert werden. Zusätzlich muss der Ordner „Resources“ kopiert werden, der dann, relativ zum Ausführungspfad, zwei Ordnerebenen weiter oben liegen muss.

V.8.3.2.3 Download der Project Notes

Als Nächstes müssen alle Project Notes heruntergeladen und ins XPS Format konvertiert werden. Dies geschieht mithilfe des Preparers, der sich unter `ProjectFlip.Preparer/bin/Release/ProjectFlip.Preparer.exe` befindet. Dieser Vorgang kann bis zu 20 Minuten dauern.

V.8.3.2.4 Ausführen der Applikation

Unter `ProjectFlip/bin/Release/ProjectFlip.exe` kann schliesslich die Applikation gestartet werden.

V.8.3.3 Weiterentwicklung

Um Project Flip 2.0 weiterentwickeln zu können, müssen folgende Schritte ausgeführt werden.

V.8.3.3.1 Entwicklungsumgebung Anforderungen

Die wichtigsten Anforderungen an die Entwicklungsumgebungen sind:

- Visual Studio 2010
- .NET, Surface 2 SDK
- Adobe Reader X (v10)
- Internetverbindung

V.8.3.3.2 SVN Import

Um die Applikation auf dem Rechner installieren zu können, muss ein SVN Checkout auf dem SVN Repository durchgeführt werden. Dazu wird ein SVN Tool wie TortoiseSVN benötigt (siehe V.4.2 Tools).

V.8.3.3.3 Ausführung

Für die Ausführung des Projekts wird das Visual Studio 2010 benötigt. Um das Projekt zu öffnen, muss die Datei `ProjectFlip.sln` im Order `code/trunk/ProjectFlip` geöffnet werden.

Nachdem das Projekt geöffnet wurde, kann der Code kompiliert werden. Es stehen zwei ausführbare Dateien zur Verfügung:

- `ProjectFlip/bin/Release/ProjectFlip.exe`
 - Startet die Benutzeroberfläche
 - Wird über `ProjectFlip.exe.config` konfiguriert
- `ProjectFlip.Preparer/bin/Release/ProjectFlip.Preparer.exe`
 - Stellt mehr Projects Note bereit, indem diese vom Zühlke Webserver heruntergeladen werden
 - Wird über `ProjectFlip.Preparer.exe.config` konfiguriert

V.8.4 Code Dokumentation

V.8.4.1 Autorenschaft

Die Autorenschaft der Code Files ist gemeinschaftlich und wird daher in den einzelnen Files nicht explizit aufgelistet.

V.8.4.2 Code Statistik

V.8.4.2.1 Testabdeckung

Die Totale Testabdeckung beträgt 83 %, wenn alle Projekte und Klassen miteinbezogen werden:

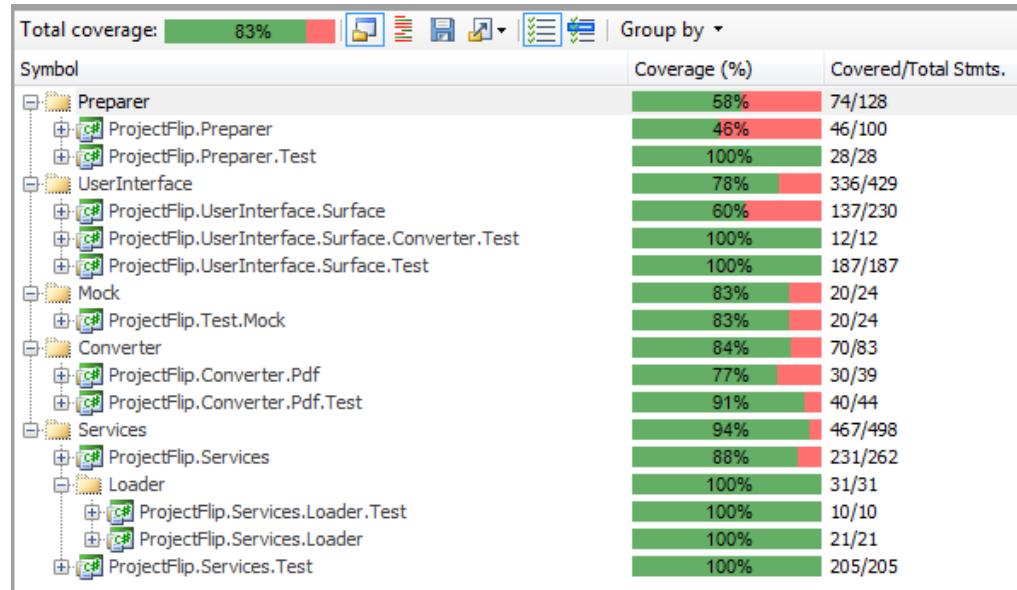


Abbildung 78 - Testabdeckung ohne ausgeblendete Klassen

Da es sich nicht lohnt oder fast nicht möglich ist, bestimmte Elemente (z.B. das Main Programm) zu testen, wurden diese Komponenten ausgeblendet. Danach sieht die Testabdeckung folgendermassen aus:

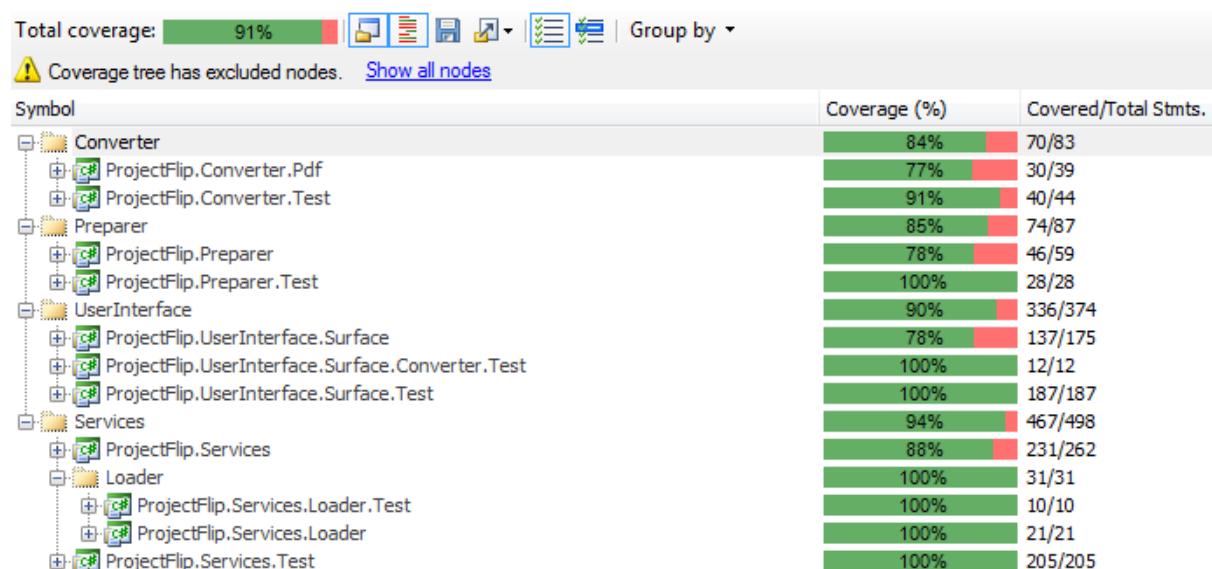


Abbildung 79 - Testabdeckung mit ausgeblendeten Klassen

Weitere Details zur Entwicklung der Testabdeckung ist dem Abschnitt V.8.2.1 Unit Tests zu entnehmen.

V.8.4.2.2 Lines Of Code (LOC)

Anhand der LOC kann abgeschätzt werden, wie umfangreich ein Projekt ist. Die LOC wurden von zwei unabhängigen Tools, für den C# Code, berechnet:

- Visual Studio: 1'129
- NDepend: 1'039

Bei beiden Werkzeugen rechnen jedoch nicht die LOC für den XAML Code aus, deshalb wurden diese manuell berechnet. Sie betragen 1'628 Zeilen, also einen durchaus grösser Teil als der C# Code.

Weitere Informationen und Grafiken sind im Unterkapitel V.8.4.3 Code Qualität zu finden.

V.8.4.3 Code Qualität

Für die Metrikanalyse des Codes wurden verschiedene Tools verwendet. Somit ist eine objektivere Bewertung des Codes möglich.

V.8.4.3.1 Visual Studio 2010

Zielsetzung war es, einen „Maintainability Index“ [microsoft11.2] von mindestens 50% zu erreichen, auf Ebene Projekt. Der „Maintainability Index“ setzt sich aus verschiedenen Kriterien zusammen und liegt zwischen 0 und 100. Ein Index zwischen 0 und 9 weist auf schlechte, ein Wert zwischen 10 und 19 auf eine moderate und zwischen 20 und 100 eine gute Wartbarkeit hin. Die Analyse ergab folgende Metriken:

Hierarchy	Maintainabi...	Cyclomatic...	Depth of ...	Class C...	Lines of C...
Converter\ProjectFlip.Converter.Interfaces (Debug)	100	3	0	0	0
Converter\ProjectFlip.Converter.Pdf (Debug)	72	14	1	9	33
Converter\ProjectFlip.Converter.Pdf.Test (Debug)	79	17	1	13	29
Mock\ProjectFlip.Test.Mock (Debug)	94	51	1	21	54
Preparer\ProjectFlip.Preparer (Debug)	77	46	1	57	105
Preparer\ProjectFlip.Preparer.Test (Debug)	81	16	1	12	22
ProjectFlip (Debug)	100	0	0	0	0
Services\Loader\ProjectFlip.Services.Loader (Debug)	76	18	1	8	24
Services\Loader\ProjectFlip.Services.Loader.Interfaces (Debug)	100	3	0	1	0
Services\Loader\ProjectFlip.Services.Loader.Test (Debug)	68	4	1	8	10
Services\ProjectFlip.Services (Debug)	80	174	2	76	295
Services\ProjectFlip.Services.Interfaces (Debug)	100	47	0	17	0
Services\ProjectFlip.Services.Test (Debug)	73	57	1	44	180
UserInterface\ProjectFlip.UserInterface.Surface (Debug)	85	113	3	49	197
UserInterface\ProjectFlip.UserInterface.Surface.Converter.Test (Debug)	82	5	1	7	11
UserInterface\ProjectFlip.UserInterface.Surface.Test (Debug)	76	39	1	36	169

Abbildung 80 - Übersicht Metriken Visual Studio

Da die Namen der Metriken im Bild abgeschnitten werden, hier eine Auflistung der Spaltennamen von links nach rechts: Maintainability Index, Cyclomatic Complexity, Depth of Inheritance, Class Coupling, Lines of Code.

Wie aus der Abbildung 80 - Übersicht Metriken Visual Studio zu lesen ist, ist der Maintainability Index bei allen Projekten der Solution über 50, was den nichtfunktionalen Anforderungen entspricht.

V.8.4.3.2 NDepend

Um noch mehr Code Metriken zu erhalten, wurde der Code auch mit NDepend analysiert. Folgende Statistiken wurden direkt von NDepend kopiert:

Application Metrics				Note: Further Application Statistics are available.			
# Lines of code : 1,039	# IL instruction : 8,894	# Exception types : 0	Third Party Usage	Percentage ...			
# Assemblies : 16	# Lines of comment : 1,908	# Attribute types : 0	# Assemblies used : 15	... code coverage N/A			
# Namespaces : 25	# Classes : 86	# Delegate types : 0	# Namespaces used : 51	... of comment : 64%			
# Types : 98	# Abstract classes : 1	# Enumeration types : 0	# Types used : 207	... of public types : 67.35%			
# Methods : 547	# Interfaces : 12	# Generic methods : 1	# Methods used : 291	... of public methods : 70.02%			
# Fields : 187	# Value types : 0	# Generic types : 2	# Fields used : 16	... of classes with public field(s) : 3.06%			
# C# source files : 92							

Abbildung 81 - Übersicht Metriken NDepend

Assemblies Metrics											
Show All <input type="checkbox"/> entries <input type="text"/> Search:											
Assemblies	# lines of code	# IL instruction	# Types	# Abstract Types	# lines of comment	% Comment	% Coverage	Afferent Coupling	Efferent Coupling	Relational Cohesion	Inst...
ProjectFlip.Services.Interfaces v1.0.0.0	0	0	9	9	-	-	-	47	17	1.11	0.27
ProjectFlip.Services.Loader.Interfaces v1.0.0.0	0	0	1	1	-	-	-	6	3	1	0.33
ProjectFlip.Services v1.0.0.0	232	2088	19	0	428	64	-	11	96	1.47	0.9
ProjectFlip.Converter.Interfaces v1.0.0.0	0	0	1	1	-	-	-	3	3	1	0.5
ProjectFlip.Converter.Pdf v1.0.0.0	36	257	2	0	62	63	-	2	23	1	0.92
ProjectFlip.Services.Loader v1.0.0.0	18	181	1	0	55	75	-	3	19	1	0.86
ProjectFlip.Preparer v1.0.0.0	91	676	10	1	191	67	-	2	73	1.9	0.97
ProjectFlip.UserInterface.Surface v1.0.0.0	194	1728	20	1	370	65	-	7	87	0.5	0.93
ProjectFlip.Test.Mock v1.0.0.0	16	306	6	0	43	72	-	8	26	0.67	0.76
ProjectFlip.Preparer.Test v1.0.0.0	21	206	2	0	80	79	-	0	14	0.5	1
ProjectFlip.UserInterface.Surface.Test v1.0.0.0	169	1001	9	0	115	40	-	0	39	0.78	1
ProjectFlip.UserInterface.Surface.Converter.Test v1.0.0.0	9	70	2	0	33	78	-	0	9	0.5	1
ProjectFlip.Converter.Test v1.0.0.0	34	239	2	0	77	69	-	0	22	1	1
ProjectFlip.Services.Test v1.0.0.0	190	1964	10	0	324	63	-	0	49	0.3	1
ProjectFlip v1.0.0.0	20	103	3	0	80	80	-	0	36	0.33	1
ProjectFlip.Services.Loader.Test v1.0.0.0	9	75	1	0	50	84	-	0	13	1	1

Abbildung 82 - Assemblies Metrics NDepend

V.8.4.3.3 Code Warnungen

V.8.4.3.3.1 Compiler Warnungen

Wie in nachfolgender Abbildung ersichtlich ist, treten beim Kompilieren des Codes keine Fehler oder Warnungen auf.

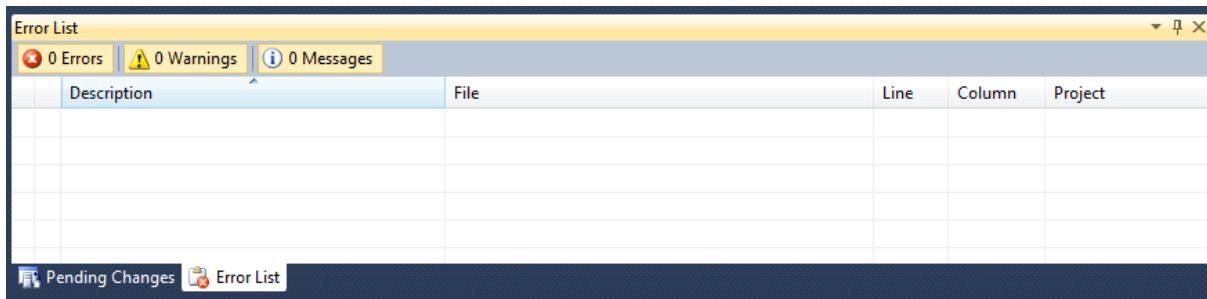
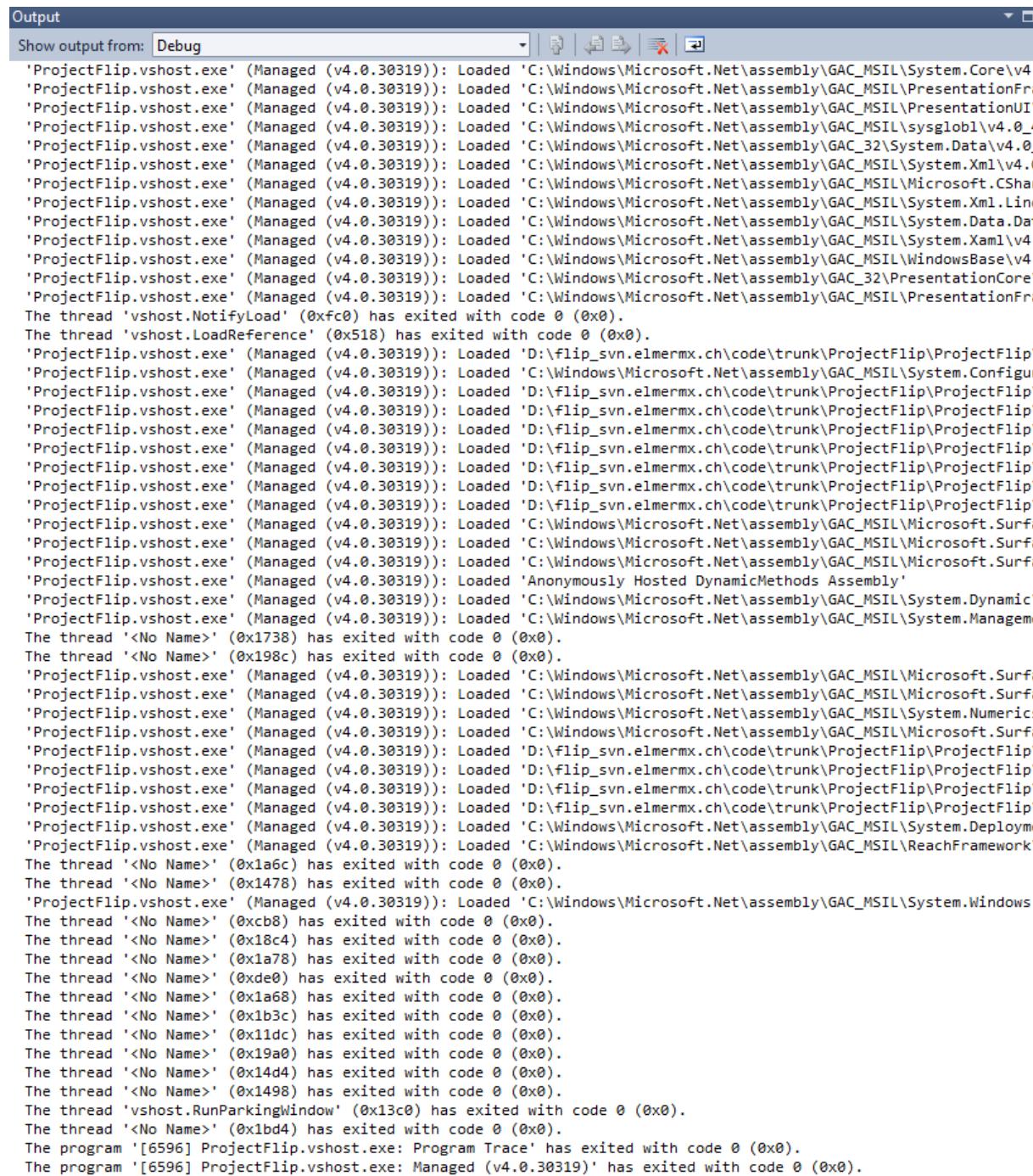


Abbildung 83 - Keine Fehler oder Warnungen beim Kompilieren

V.8.4.3.3.2 Laufzeitwarnungen

Wird die Applikation ausgeführt und alle User Stories durchgeführt, so treten keine Warnungen auf.



```

Output
Show output from: Debug

'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Core\v4.0_4
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\PresentationFra
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\PresentationUI'
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\sysglobl\v4.0_4
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\System.Data\v4.0_
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml\v4.0_
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.CShar
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xml.Linq
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Data.Dat
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Xaml\v4.
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\WindowsBase\v4.
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_32\PresentationCore'
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\PresentationFra
The thread 'vhost.NotifyLoad' (0xfc0) has exited with code 0 (0x0).
The thread 'vhost.LoadReference' (0x518) has exited with code 0 (0x0).
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'D:\flip_svn.elmermx.ch\code\trunk\ProjectFlip\ProjectFlip
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Configur
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'D:\flip_svn.elmermx.ch\code\trunk\ProjectFlip\ProjectFlip
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'Anonymous Hosted DynamicMethods Assembly'
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Dynamic
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Manageme
The thread '<No Name>' (0x1738) has exited with code 0 (0x0).
The thread '<No Name>' (0x198c) has exited with code 0 (0x0).
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Numerics
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.Surfa
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'D:\flip_svn.elmermx.ch\code\trunk\ProjectFlip\ProjectFlip
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Deployme
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\ReachFramework'
The thread '<No Name>' (0x1a6c) has exited with code 0 (0x0).
The thread '<No Name>' (0x1478) has exited with code 0 (0x0).
'ProjectFlip.vhost.exe' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Windows.
The thread '<No Name>' (0xcb8) has exited with code 0 (0x0).
The thread '<No Name>' (0x18c4) has exited with code 0 (0x0).
The thread '<No Name>' (0x1a78) has exited with code 0 (0x0).
The thread '<No Name>' (0xde0) has exited with code 0 (0x0).
The thread '<No Name>' (0x1a68) has exited with code 0 (0x0).
The thread '<No Name>' (0xb3c) has exited with code 0 (0x0).
The thread '<No Name>' (0x11dc) has exited with code 0 (0x0).
The thread '<No Name>' (0x19a0) has exited with code 0 (0x0).
The thread '<No Name>' (0x14d4) has exited with code 0 (0x0).
The thread '<No Name>' (0x1498) has exited with code 0 (0x0).
The thread 'vhost.RunParkingWindow' (0x13c0) has exited with code 0 (0x0).
The thread '<No Name>' (0xbd4) has exited with code 0 (0x0).
The program '[6596] ProjectFlip.vhost.exe: Program Trace' has exited with code 0 (0x0).
The program '[6596] ProjectFlip.vhost.exe: Managed (v4.0.30319)' has exited with code 0 (0x0).

```

Abbildung 84 - Keine Warnungen bei der Ausführung aller User Stories

V.8.4.4 Coding Standards

Die Codings Standards, welche für dieses Projekt gültig sind, wurden mehrheitlich vom Resharper übernommen.

Nachfolgend sind die unveränderten Standards mittels Bildausschnitten dokumentiert.

V.8.4.4.1 C# Namenskonventionen

Folgende Namenskonventionen wurden verwendet:

Entity kinds	Preview
Types and namespaces	UpperCamelCase
Interfaces	IUpperCamelCase
Type parameters	TUpperCamelCase
Methods, properties and events	UpperCamelCase
Local variables	lowerCamelCase
Local constants	lowerCamelCase
Parameters	lowerCamelCase
Fields (not private)	UpperCamelCase
Instance fields (private)	_lowerCamelCase
Static field (private)	_lowerCamelCase
Constant fields (not private)	UpperCamelCase
Constant fields (private)	UpperCamelCase
Static readonly fields (not private)	UpperCamelCase
Static readonly fields (private)	UpperCamelCase
Enum members	UpperCamelCase
All other entities	UpperCamelCase

Abbildung 85 - Naming Style

V.8.4.4.2 Formatierungsstil

V.8.4.4.2.1 Braces Layout

Geschweifte Klammern befinden sich auf einer neuen Zeile.

Ausnahme: geschweifte Klammern, die nichts enthalten. (Bsp: internal Class C {}).

```
namespace N
{
    internal interface I
    {
        void foo();
    }

    internal class C {}
}
```

Abbildung 86 - Braces Layout

V.8.4.4.2.2 Line Breaks und Wrapping

Lange Zeilen (>120 Zeichen) werden umgebrochen.

```
string output = string.Format(CultureInfo.InvariantCulture,
    "{0:yyyy-MM-dd} {1}", date, message);
```

Abbildung 87 - Line Breaks and Wrapping

V.8.4.4.3 CleanUp

Für das CleanUp des Codes wurden folgende Einstellungen vorgenommen:

C#	
Arrange 'this' qualifier	<input checked="" type="checkbox"/>
Remove code redundancies	<input checked="" type="checkbox"/>
Use auto-property, if possible	<input checked="" type="checkbox"/>
Make field read-only, if possible	<input checked="" type="checkbox"/>
Use 'var' in declaration	
Replace direction	Can change type usage to 'var'
foreach iterator declaration style	Always use 'var'
Local variable declaration style	Always use 'var'
Update file header	<input checked="" type="checkbox"/>
Optimize 'using' directives	
Optimize 'using' directives	<input checked="" type="checkbox"/>
Embrace 'using' directives in region	<input checked="" type="checkbox"/>
Region name	
Shorten qualified references	<input checked="" type="checkbox"/>
Reformat code	<input checked="" type="checkbox"/>
Reformat embedded XML doc comments	<input type="checkbox"/>
Reorder type members	<input checked="" type="checkbox"/>

Abbildung 88 - Code Clean Up Einstellungen

V.8.4.5 Dokumentation Quellcode

Zur Dokumentation des Quellcodes wurde Ghost Doc verwendet. Dadurch war die Dokumentation einiges einfacher, da sinnvolle Standardkommentare generiert werden, die bei Bedarf erweitert werden können. Durch die geschriebenen XML Kommentare ist es auch möglich, eine Dokumentation zu generieren.

Es wurden alle Interfaces, sowie Methoden oder Properties, welche mit Hilfe eines Kommentars besser verstanden werden, kommentiert. Ausgenommen davon sind die Tests und die XAML Dateien. Durch die Kommentare sind der Programmcode und besonders komplexe Methoden für Entwickler leichter verständlich.

V.8.5 Code Reviews

Um die Code Qualität zu gewährleisten, wurden im Team immer wieder Code Reviews durchgeführt. Diese wurden vor Abschluss jedes Sprints partnerschaftlich durchgeführt. Im Code wurden keine Vermerke für diese Reviews angebracht.

Zusätzlich zu den internen Reviews wurden auch Code Reviews mit den externen Personen Christian Moser und Michael Gfeller durchgeführt.

Datum	Review	Externe Person
04.11.2011	Review Sprint 4	Christian Moser, Zühlke Engineering AG
25.11.2011	Review Sprint 5	Christian Moser, Zühlke Engineering AG
09.12.2011	Review Sprint 6	Michael Gfeller, IFS HSR
16.12.2011	Review Projektabgabe	Michael Gfeller, IFS HSR

Tabelle 14 - Code Reviews mit externen Personen

V.8.5.1.1 04.11.2011

Review mit: Christian Moser

Anwesend: Christian Moser, Christina Heidt, Delia Treichler, Lukas Elmer

Besprochene Punkte:

- Toucherkennung bei 3-D Elementen in der Übersicht (evt. Masterstudent der unterstützen kann). Ansonsten mit Transformation arbeiten anstatt 3-D.
- Arbeiten entweder mit 3-D oder 2-D Modell. Elemente müssten 3-D aufgebaut werden (Licht, Texturen etc.) oder mit Transformationen im 2-D Bereich.
- Am Anfang einfache Lösung mit Scrollbalken, bei Zeit Ausbau mit 3-D. Wechsel von Übersicht in Detailansicht ist wichtiger.

V.8.5.1.2 25.11.2011

Review mit: Christian Moser

Anwesend: Christian Moser, Christina Heidt, Delia Treichler, Lukas Elmer

Besprochene Punkte:

- Um die Animationen zu gestalten, wurden VisualStateGroups eingesetzt. Diese wurden im Team besprochen und im Expression Blend erstellt.
- Um spezielle Events abzufangen können, werden statische Klassen eingesetzt werden. So benötigt man keinen Code Behind.
- Es kann mit wenig Code Behind gearbeitet werden, solange sich dieser nicht auf die Business Logik sondern nur auf das User Interface auswirkt.
- Die Animationen werden nicht richtig angezeigt, da die Rechenleistung für das Laden des XPS Dokumentes verwendet wird. Aus diesem Grund wird das Dokument erst nach der Animation geladen.

V.8.5.1.3 09.12.2011

Review mit: Michael Gfeller

Anwesend: Michael Gfeller, Christina Heidt, Delia Treichler, Lukas Elmer

- Dadas XAML des OverviewWindows zu umfangreich ist, sollen die Styles in ein Styles.xaml ausgelagert werden. Diese können dann mithilfe eines ResourceDirectory ins XAML eingebunden werden (Merged Resource Dictionaries).
- Images sind nicht freezed → dadurch könnten Memory Leaks entstehen
- ScrollToTopBehavior: DependencyPropertyDescriptor ist statisches Konstrukt (ItemsSourceProperty): prüfen, dass es sich abmeldet, sonst könnte hier ein Memory Leak entstehen.
- Die einzelnen Klassen sollen aufgeräumt werden: Verschiedene Regionen von oben nach unten: Deklarationen, Properties, Konstruktor, dann private Variablen und Methoden.
- Absoluter Pfad zu Adobe Reader soll in ein Konfigurations File ausgelagert werden.

- Installer fehlt, da aber keine Hardware verfügbar ist, kann dieser noch nicht erstellt werden. In Dokumentation soll erklärt werden, was alles noch getan werden muss, um Projekt auf dem Surface 2 zu benutzen.
- Code dokumentieren für public Methoden/Properties/Klassen, falls sich dies bei Abschnitt lohnt (speziell wenn nicht klar ist, um was es sich handelt, z.B. bei Preload() im ProjectNote Model).

Beschlüsse:

- Da nur ein ViewModel existiert, wird kein separates Projekt für die ViewModels erstellt.

V.8.5.1.4 16.12.2011

Review mit: Michael Gfeller

Anwesend: Michael Gfeller, Lukas Elmer

- Laufzeit Warnungen: Beim Binding gibt es Laufzeit Warnungen und zwar in der Detailansicht beim Navigieren von einer Project Note zur nächsten. Dies wird durch das Binding verursacht und könnte daran liegen, dass auf null gebindet wird. Die Behebung dieser Warnungen würde jedoch zu viel Zeit in Anspruch nehmen.
- Im GravatarsViewModel wird eine IList verwendet. Besser wäre hier eine ListView, da durch das automatische Binding / Konvertieren ein Memory Leak entstehen könnte.
- Durch die Installation des WPF Performance Kit könnte man die Ressourcen besser überwachen und beobachten.
- GravatarsViewModelTest: Dieser Test bringt nur wenig Mehrwert.
- Die Konfiguration app.config in den unteren Projekten (Bsp. PdfConverter) bringt nichts, da die Datei nicht ins Verzeichnis mit den ausführbaren Dateien kopiert wird.

```
System.Windows.Data Error: 4 :  
Cannot find source for binding with reference  
'RelativeSource FindAncestor,  
AncestorType='System.Windows.Controls.ItemsControl',  
AncestorLevel='1''.BindingExpression:Path=HorizontalContentAlignment;  
DataItem=null;  
target element is 'SurfaceListBoxItem' (Name='');  
target property is 'HorizontalContentAlignment'  
(type 'HorizontalAlignment')
```

Abbildung 89 - Laufzeitwarnungen

Beschlüsse:

- Aufgrund Zeitdrucks ist es nicht möglich, alle beanstandeten Punkte umzusetzen. Deshalb werden die Laufzeit Warnungen, die sehr wahrscheinlich durch das Binding auf Null-Objekte ausgelöst werden, nur sehr kurz untersucht und, falls nicht einfach lösbar, ignoriert.
- Die Konfiguration wird noch kopiert, damit Einstellungen vorgenommen werden können.
- Die anderen Punkte werden berücksichtigt, falls noch Zeit bleiben sollte.

Project Flip 2.0

Studienarbeit

V.9 Schlussfolgerung

V.9.1 Dokumentinformationen

Datum	Version	Änderung	Autor
14.12.2011	1.0	Erste Version des Dokuments	cheidt
15.12.2011	1.1	Review	dreichl
21.12.2011	1.2	Korrekturen	dreichl

V.9.2 Ergebnisse

Folgende Anforderungen wurden zu Beginn der Arbeit von der Zühlke Engineering AG gestellt (die Aufgabenstellung ist im VII Anhang, A zu entnehmen)

- **Das PF2 stellt Project Notes ansprechend dar und erlaubt eine intuitive Navigation. (Prio 1)**
Die Project Notes können dank ihrem vektorisierten Format beliebig vergrössert werden und sind daher gut auf einem Bildschirm lesbar. Die Darstellung der einzelnen Project Notes in der Übersicht orientiert sich stark am Design der effektiven Project Notes, der Bildausschnitt dient zusätzlich zur einfacheren Unterscheidung der einzelnen Elemente in der Liste.
Die Navigation orientiert sich an klassischen, einfachen und wohlbekannten Konzepten und ist daher einfach verständlich. Objekte, mit welchen interagiert werden kann, werden optisch von der übrigen Applikation abgehoben.
- **Das PF2 erlaubt intuitive Suche und Filterung der Project Notes (Volltext, Branche, Kunde, Technologie etc.) (Prio 1)**
Die Filterung erfolgt nach den Informationen, welche aus den zahlreichen Metadaten extrahiert wurden. Um den Nutzer aber nicht mit einer riesigen Anzahl von Begriffen zu überfordern, wurden verschiedene Begriffe zu einem Überbegriff zusammengefasst. Diese Angaben können einfach über eine Mapping Datei angepasst werden.
Die Filterkriterien können aus einer Liste ausgewählt und dem Filter hinzugefügt werden.
Die Volltextsuche wurde nicht umgesetzt, da diese als weniger wichtig eingestuft wurde im Vergleich zu anderen Anforderungen.

Nach Beginn des Projektes wurde bekannt, dass sich der Erscheinungszeitpunkt der Hardware verschiebt und diese erst ab Februar des kommenden Jahres 2012 verfügbar sein wird. Dadurch konnten die nachfolgenden Anforderungen nicht wie gewünscht umgesetzt werden.

- **Das PF2 greift die Project Notes direkt aus dem Zühlke Intranet ab („No Maintenance“) (Prio 2)**
Die Anbindung an den Sharepoint der Zühlke Engineering AG wurde nicht umgesetzt. Dafür wurde eine Server-Anbindung realisiert, über diese die Project Notes automatisch heruntergeladen und in ein XPS-Dokument umgewandelt werden können.
Für die Weiterentwicklung des Projektes steht ein Interface zu Verfügung, welches der Implementation der Sharepoint-Anbindung dienen soll.
- **Das PF2 erlaubt das Verschicken von Project Notes (Email / Post). (Prio 2)**
In der Vorstudie wurde dieser Punkt in Betracht bezogen und die Möglichkeiten für dessen Umsetzung dokumentiert. Dazu wurden auch entsprechende User Stories definiert, aber nicht umgesetzt.
- **Das PF2 erlaubt eine Registrierung. Diese kann über Visitenkarten-Scan, Handy oder andere Mechanismen stattfinden. (Prio 2)**
In der Vorstudie wurde dieser Punkt in Betracht bezogen und die Möglichkeiten für dessen Umsetzung dokumentiert. Dazu wurden auch entsprechende User Stories definiert, aber nicht umgesetzt.

Die Teammitglieder sammelten durch die Entwicklung der Project Flip 2.0 - Applikation erste, äusserst lehrreiche Erfahrungen mit WPF, .NET und dem Surface 2.0 SDK. Trotz Einarbeitungszeit ist es gelungen, einen funktionstüchtigen und ansprechenden Prototyp zu entwickeln. Zudem konnten die Anforderungen mit der höchsten Priorität umgesetzt werden.

Die Qualität der Anwendung wurde während der Entwicklung immer wieder durch Testpersonen, welche den evaluierten Nutzerprofilen entsprechen, verifiziert.

Abschliessend ist zu bemerken, dass Project Flip 2.0 eine solide, gut ausbaubare Grundlage für die Weiterentwicklung zur vollwertigen Applikation bildet. Ferner wurde aufgezeigt, dass sich die Projekte der Zühlke Engineering AG durch eine Surface 2 Applikation bestens visualisieren lassen.

V.9.2.1 Vergleich zur bestehenden Lösung

Zurzeit stehen die Project Notes im Wartebereich in Papierform zur Verfügung. Die Nutzung einer Surface Applikation für deren Präsentation bietet die folgenden Vorteile:

- Auf technikbegeisterte Personen wirkt eine neue Technologie immer anziehend. So können Personen, die zuvor keinen Anreiz sahen, sich mit den Projekten der Zühlke Engineering AG auseinanderzusetzen, dadurch motiviert werden, dies doch zu tun.
- Es sind immer alle bestehenden Projekte verfügbar.
- Es ist einfach, nach spezifischen Inhalten oder Themen zu suchen.
- Durch die Anzeige der Projektinformationen (Metadaten), können Nutzer auf andere, neue Themen und damit verknüpfte Projekte aufmerksam gemacht werden.
- Eine einfache und ansprechende Applikation animiert den Benutzer, diese zu entdecken und gleichzeitig etwas über die Firma zu lernen.
- Die Zühlke Engineering AG ist unter anderem im Bereich Software-Engineering tätig. Durch eine Surface 2 Applikation kann sie beweisen, dass auch neue Technologien mit Leichtigkeit gemeistert werden.
- Den Kunden kann ein mögliches Einsatzgebiet des Surface 2, nämlich als Ausstellungsobjekt, aufgezeigt werden.

V.9.3 Ausblick

Bevor Project Flip 2.0 effektiv eingesetzt werden kann, müssen nachfolgend aufgelistete Anforderungen umgesetzt werden:

- Die Anbindung an den Sharepoint muss umgesetzt werden. Dazu steht ein Interface zu Verfügung, welches dementsprechend implementiert werden muss.
- Die Verwendung von Gesten soll umgesetzt werden.
Im Usability Test zeigte sich, dass die Testpersonen intuitiv Gesten benutzen wollten, um
 - das Filterkriterium in und wieder aus dem Filter ziehen zu können
 - die Project Note zu vergrößern und verkleinern
- Die Performance ist zu verbessern. Es dauert einen Augenblick, um die veränderte Liste von Project Notes anzuzeigen, wenn ein Filter gesetzt oder wieder entfernt wird. Eine Performance-Verbesserung könnte durch das Virtualisieren des Panels in der Übersicht erreicht werden.
- Wie in der Aufgabenstellung (die Aufgabenstellung ist im VII Anhang, A zu entnehmen) ersichtlich ist, ist eine Volltextsuche zu implementieren.
- Die Farben der Applikation können auf dem Surface 2 eine ganz andere Wirkung haben als an einem normalen Monitor. Diese sind daher zu überprüfen.
- Eine Project Note kann auf ein bestimmtes Maximum vergrößert werden. Diese Angabe muss ebenfalls auf der Hardware verifiziert werden.
- Für einen besseren Programmfluss müssen die bestehenden Animationen generell erweitert werden.
Zu priorisieren sind
 - Animation für das Herein- und Herausziehen des Filterkriteriums
Wird im Menü ein Filterkriterium angetippt wird, so soll dieses in einen freien Slot des Filters fliegen, um dem Nutzer anzuzeigen, dass die Filterkriterien auch in den Slot gezogen werden können. (Siehe Auflistungspunkt 2, Verwendung von Gesten)
 - Die Projektinformationen (Metadaten) sollen bei der Auswahl einer Project Note aus der Übersicht mit einer Animation erscheinen. Zudem soll auch beim Navigieren durch die Project Notes in der Detailansicht eine Animation für die Anzeige der Projektdaten verwendet werden. So ist für den Nutzer ersichtlich, dass die Projektinformationen pro Project Note ändern.
 - Wird eine Project Note in der Übersicht ausgewählt, so soll sich diese von ihrer Position in der Liste zur Mitte des Bildschirms hin vergrößern. Dies macht dem Benutzer deutlich, welche Project Note angetippt wurde. (Die Animation, welche in diesem Prototyp umgesetzt ist, startet die Vergrößerung der Project Note immer in der Mitte des Bildschirmes, unabhängig von der Position der ausgewählten Project Note.)
- Das Betriebskonzept steht in Grundzügen zur Verfügung und ist auszuarbeiten.
- Die Installationsanleitung steht in Grundzügen zur Verfügung und ist auszuarbeiten.

Project Flip 2.0

Studienarbeit

VI. Projekt
Retrospektive

VI.1 Dokumentinformationen

Datum	Version	Änderung	Autor
18.12.2011	1.0	Erste Version des Dokuments	lelmer
19.12.2011	1.1	Persönlicher Bericht	cheidt
20.12.2011	1.2	Aufwandanalyse, Diagramme, Tabellen	lelmer
20.12.2011	1.3	Persönlicher Bericht	dreichl
21.12.2011	1.4	Review	dreichl

VI.2 Methoden und Technologien

Zu Beginn von Project Flip 2.0 wurde eine grobe Projektplanung im Redmine (Projektplanningstool und Ticketingsystem) gemacht, in der die wichtigsten Aspekte beschrieben wurden und alle Kalenderdaten festgehalten wurden.

Während des Projektes wurde dann stark agil nach Scrum gearbeitet. Die 14 Wochen, welche das Projekt dauerte, wurden in 7 Sprints zerlegt, in denen man sich auf gewisse Punkte konzentrierte.

Zu Beginn jedes Sprints wurde die Planung für den Sprint durchgeführt, was sich als riesigen Vorteil erwies, denn so konnten die Anforderungen viel genauer aufgefasst werden.

Konnte ein Feature, welches für einen bestimmten Sprint geplant war, nicht vor dessen Ende fertig implementiert werden, wurde es in den nächsten Sprint verschoben. So konnten die Abgabetermine eines Sprints immer eingehalten werden.

Da Usability für das Projekt ein sehr wichtiger Punkt war, wurde benutzerorientiert entwickelt. Darum wurde vor dem Programmieren der Lösung viel Zeit in die Erstellung von Personas, Szenarien und einem Papierprototypen investiert und viele neue Lösungen ausprobiert, was sehr interessant war.

Um die Codequalität zu gewährleisten und die Applikation einfacher weiterentwickeln zu können, wurde sie in verschiedene kleine Teilprojekte unterteilt. Diese wurden dann auch mit Unit Tests stark überdeckt, zwischen 80% und 95%. Dies erwies sich als grossen Vorteil, da so schnell Probleme erkannt wurden, wenn etwas nicht wie gewünscht funktionierte.

Da die Microsoft Welt für alle Teammitglieder noch unbekannt war, waren die Schätzungen nicht immer einfach. Das war ein weiterer Grund, um Scrum einzusetzen, denn so konnte der Umfang bei jedem Sprint angepasst werden und alle Mitglieder wurden dazu gezwungen, bei Problemen externe Hilfe zu suchen.

Die .NET / WPF Technologie ist zweifellos sehr gross, mächtig und interessant. Dies hat sich während des Projektes aber nicht immer nur positiv auf die Arbeit ausgewirkt, denn für gewisse Probleme wurde viel Zeit investiert, um ein kurzen und einfachen Befehl zu finden. WPF war auch in Bezug auf Performance nicht immer einfach.

Leider war der Zeitrahmen der Arbeit beschränkt und es konnte nicht alles so implementiert werden, wie wir es gerne gewollt hätten und es uns beim Papierprototyp vorgestellt hatten. Zum Beispiel hatten wir keine Zeit mehr, die Gesten zu implementieren und auch für die Animationen stand nicht so viel Zeit zur Verfügung, wie wir es gerne gehabt hätten.

Sehr schön war das Gefühl, ein starkes, motiviertes Team mit viel Wissen hinter sich zu haben. So konnten wir während der vollen Projektdauer bei Unklarheiten um Rat fragen, ohne viel Zeit zu verlieren.

Ganz speziell unterstützt haben uns Markus Stolze (IFS, HSR), Christian Moser und Marco Balzarini (Zühlke Engineering AG). An dieser Stelle einen ganz herzlichen Dank an diese Personen.

VI.3 Persönliche Berichte

VI.3.1 Lukas Elmer

Gegen Ende des 4. Semesters mussten sich die Studenten für eine Studienarbeit entscheiden. Da ich bereits einiges an Erfahrung in Skriptsprachen und Open Source hatte, entschied ich mich für eine Microsoft Technologie, um auch diese Schiene kennen zu lernen. Usability sehe ich, speziell in der Zukunft, als ein sehr wichtiges Differenzierungsmerkmal zwischen zwei Softwarelösungen. So interessiere ich mich auch für ein benutzerzentriertes Vorgehen und war so bei Markus Stolze an der richtigen Adresse.

Das Projekt startete, ungleich der meisten anderen Studienarbeiten, bereits vor dem Semesterstart. Da ich mir unter Zühlke Engineering AG noch nicht viel vorstellen konnte, ging ich auch ohne grosse Erwartungen an das Kickoff Meeting, an dem ich dann sehr positiv überrascht wurde. Christian Moser und Marco Balzarini von der Zühlke Engineering AG waren von Beginn an schon sehr motiviert und zogen diese Motivation auch bis ans Ende des Projektes durch, was ich extrem cool finde. Nach einer grundlegenden Einführung in die Microsoft Technologien, speziell in C#, WPF und das Visual Studio, wurden wir immer wieder von Christian unterstützt und auch motiviert durch seine positive Kritik.

Ein sehr interessanter Aspekt bei der Anwendung von Scrum in diesem Projekt ist für mich, dass das Produkt zwar noch nicht fertig ist, jedoch ein voll funktionsfähiger, stabiler und getester Prototyp entwickelt werden konnte. Diesen könnte man bereits so benutzen, wie er jetzt ist, es ist einfach noch nicht die ganze Funktionalität vorhanden.

In einem nächsten Projekt würden wir wieder zweiwöchentliche Sprints ansetzen, also etwa 32h pro Person pro Sprint. Wenn also zu 100% an einem Projekt gearbeitet wird, können bestimmt auch wöchentliche Sprints durchgeführt werden.

Im Team lief zu Beginn der Arbeit auch alles sehr gut, denn alle Teammitglieder waren sehr motiviert und engagiert. Das änderte sich leider ein wenig zum Negativen, da ich einerseits sehr viel zu tun hatte, andererseits weil gewisse Probleme sehr knifflig zu lösen waren und so die Frustration wuchs. Zusätzlich hatte ich auch noch einen anderen Stundenplan als Christina und Delia, was uns natürlich zusätzlich auseinander drängte. Auch wollten wir das Challenge Projekt zusammen absolvieren, doch leider konnte ich das Modul nicht noch einmal besuchen, weil ich es bereits ein Semester vorher abgeschlossen hatte. Dies alles führte dazu, dass ich nur zu etwa 70% der Zeit anwesend war. Trotzdem trug ich natürlich meinen Teil zur Arbeit bei und erledigte meine Aufgaben zuverlässig. Problematisch war wahrscheinlich, dass die Slots, in denen man zusammen gearbeitet wird, nicht genau genug definiert wurden.

Für die Bachelor Arbeit bin ich mir sicher, wieder mit dem gleichen Team arbeiten zu wollen, da die einzelnen Teammitglieder zuverlässig und motiviert sind. Es muss aber klar definiert werden, wer wann anwesend sein wird.

Etwas enttäuschend war, dass der Surface 2 nicht rechtzeitig geliefert werden konnte. Deshalb konnte die Applikation nie mit der richtigen Hardware getestet werden und gewisse Features mussten weggelassen werden, wie zum Beispiel die Erkennung von Objekten auf der Bildschirmoberfläche.

Technisch war das Projekt speziell interessant, da die ganze Microsoft Technologie neu für mich war. Es gibt viele lobenswerte Dinge, wie z.B. die C# Syntax, das MVVM Pattern, wodurch sich auch die GUI Funktionalität sehr schön testen lässt. Oder die Kapselung der einzelnen Projekte, die so auf einfache Art ausgetauscht werden können. Es sind aber auch einige böse Überraschungen aufgetreten – so waren die Animationen und die Touch Gesten viel schwieriger zu implementieren als eingeschätzt. Doch allgemein machen die eingesetzten Technologien Freude.

Zusammenfassend würde ich das Projekt als erfolgreich kennzeichnen, da wir sehr viel gelernt haben und auch die Zühlke von der Arbeit profitieren kann. Sobald das Projekt auf dem Surface 2 deployed ist, freue ich mich, die Applikation auf der richtigen Hardware zu erleben.

VI.3.2 Christina Heidt

Die eigentliche Einarbeitung für das Projekt begann schon vor Anfang des Semesters. Wir wurden für ein erstes Kennenlernen und für eine Einführung in das Projekt von Marco Balzarini und Christian Moser zur Zühlke Engineering AG eingeladen. Der Empfang war sehr herzlich und das persönliche Interesse an uns Studenten empfand ich als sehr positiv. Da wir alle noch keine Erfahrungen mit .NET und WPF gemacht hatten, führte uns Christian Moser einen Nachmittag lang in die grundlegenden Kenntnisse ein. Auch im weiteren Projektverlauf nahm er sich immer wieder Zeit, die Applikation mit uns zu prüfen und gemeinsam zu verbessern. Dieses Verhalten finde ich sehr lobenswert, da es zeigt, dass die Zühlke Engineering AG nicht nur an den Resultaten der Arbeit interessiert ist, sondern auch gewillt ist, dafür Zeit zu investieren. Auch die Betreuung durch Markus Stolze war sehr partnerschaftlich und Kritiken stets konstruktiv. Die ausgezeichnete Leistung von beiden Seiten wirkte auf mich sehr motivierend.

Die Gruppenarbeit funktionierte zu Beginn reibungslos. Im Verlauf des Semesters verschlechterte diese sich jedoch zunehmend. Dies war einerseits auf die vielen zusätzlichen Arbeiten neben der Studienarbeit – 2 Miniprojekte und das Besuchen des Faches Challengeprojekte – zurückzuführen. Andererseits entschied sich Lukas Elmer in diesem Sommer dazu, neben der Schule noch zu arbeiten, was zu einer Überlastung seinerseits führte. Daraus ergab sich, dass oftmals nur zwei Teammitglieder zur gleichen Zeit arbeiteten. Durch den begrenzen Zeitrahmen sahen diese sich oft dazu gezwungen, Entscheidungen alleine zu treffen. Diese sollten aber immer mit allen Mitgliedern getroffen werden, damit niemand bei der Entscheidungsfindung ausgeschlossen wird. Die Erfahrungen dieses Semesters werden auch zu kleinen Umstrukturierungen für die Bachelorarbeit führen, um die Zusammenarbeit zu optimieren.

Der Fokus auf der grafischen Oberfläche und einer einfachen Interaktion mit dem System waren der Grund, weshalb mich dieses Projekt besonders angesprochen hatte. Eine Surface 2 Applikation besitzt auch immer eine spielerische Komponente, welche nicht bei vielen Arbeiten vorkommt und einen zusätzlichen Reiz ausmachte. Dank der Einführung von Christian Moser und das Besuchen des Faches Microsoft Technologien, konnte ich mich schnell in WPF und .NET einarbeiten. Auch der Umgang mit dem Surface 2 SDK war anfangs relativ simpel. Besonders interessant fand ich auch die Interviews und die Prototyperstellung.

Die Realisierung der Animationen hatte ich mir zu Beginn einfacher vorgestellt. Auch hatte ich nicht erwartet, Probleme mit der Erkennung der Touch Inputs zu bekommen. Dies führte dazu, dass anfängliche Ideen schliesslich anders realisiert werden mussten.

Abschliessend ist zu sagen, dass trotz einiger Probleme ein ansehnliches Ergebnis erreicht werden konnte. Auch bin ich sehr neugierig, wie das Endergebnis schliesslich auf dem Surface 2 wirken wird.

VI.3.3 Delia Treichler

Ein erstes Treffen mit unserem Projektpartner, der Zühlke Engineering AG, fand bereits vor Semesterstart statt. Christian Moser und Marco Balzarini luden uns nach Schlieren in ihre Firma ein, für eine Einführung in die Problemstellung und ein erstes Kennenlernen.

Da unser Team noch keine Erfahrung mit .NET und WPF hatte, nahm sich Christian Moser noch vor Projektbeginn extra einen Nachmittag Zeit für uns, für eine Einführung in WPF an der HSR.

Um Ideen zu sammeln, wie die Applikation am Ende des Projektes aussehen sollte, führten wir als Team einen Creative Workshop durch. Für mich war das etwas Neues. Unsere erste Ideenskizze gefiel mir sehr gut. Ich war überzeugt, dass sie unsere Lösung sein würde. Trotzdem arbeiteten wir noch an weiteren Varianten.

Anschliessend galt es, die von uns favorisierte Idee durch einen Papierprototyp zu veranschaulichen. Doch bereits beim Erstellen unseres Papierprototyps entstanden Probleme und später, beim Durchspielen des Probeszenarios, kam es zu einer völligen Verwirrung.

So entwarfen wir, auch dank des Feedbacks unserer Testpersonen, einen zweiten und schliesslich den endgültigen Papierprototyp.

Nachdem die Implementation der Software bis über die Projektmitte hin problemlos verlaufen war, hätte ich nicht gedacht, dass uns das Programmieren der Animationen und Gesten zum Schluss solche Mühe bereiten würden.

Die Zusammenarbeit im Team war eine Herausforderung. Drei ganz verschiedene Persönlichkeiten mit je einem eigenen Arbeitsstil mussten sich immer wieder finden. Auch der Wissensstand der Teammitglieder ist unterschiedlich, was für die Arbeit und das gemeinsame Wirken sowohl positive wie auch negative Einflüsse hatte. Die Teamarbeit verlief nicht immer reibungslos.

Doch alle haben ihre Stärken, die zum Gelingen des Projektes beigetragen haben.

Christian und Marco haben uns von Beginn an unterstützt. Die Zusammenarbeit war immer sehr herzlich, was ich sehr geschätzt habe. Christian hat sich auch während des Projektes immer wieder Zeit genommen, uns bei Fragen und Unklarheiten zu helfen und den Stand der Applikation zu validieren.

Unser Betreuer, Markus Stolze, hat mit seiner guten Führung (durch die wöchentlichen Reviewsitzungen) und seinen Anregungen zum Gelingen des Projektes beigetragen.

In diesem Projekt habe ich das erste Mal mit .NET und WPF gearbeitet und sogleich Gefallen daran gefunden. Ich bin sehr gespannt, wann die Software bei der Zühlke Engineering AG auf dem Surface in Betrieb sein wird und freue mich, unsere Applikation hoffentlich bald einmal real testen zu können.

VI.4 Aufwandsanalyse

Das Projekt lief über ein Semester von 14 Wochen. Für das Modul Semesterarbeit werden 8 ETCS-Punkte pro Student vergeben. Pro ECTS-Punkt wird mit einem Aufwand von 30 Stunden gerechnet. Also standen für die Durchführung des Projektes $3 * 8 * 30 = 720$ Stunden zur Verfügung. Die ergab in diesem Projekt pro Sprint (SP 1 - SP 7) knapp 103 Stunden.

Zu den sieben Sprints des Projektes gibt es einen zusätzlichen Sprint 0, in dem die Teammitglieder die Zühlke Engineering AG kennen lernen und sich bereits ein wenig mit den neuen Technologien .NET und WPF vertraut machen konnten.

Der in das Projekt investierte Aufwand war in jedem Sprint etwa gleich gross und konstant etwas über der empfohlenen Zeit von 103 Stunden.

In den folgenden Abschnitten werden Auswertungen über die Sprints und die Mitglieder analysiert.

VI.4.1 Sprints

Die folgende Tabelle zeigt die geplanten und die tatsächlich benötigten Stunden für das Projekt. Es ist zudem ersichtlich, wie gross der effektive Aufwand von der Schätzung abweicht.

Sprint	Geplante Stunden	Aufwand in Stunden	Abweichung in Stunden	Absolute Abweichung in Stunden
0	19	19.00	0	0
1	131	126.50	-4.5	4.5
2	92	111.00	19	19
3	123.25	103.50	-19.75	19.75
4	125.5	119.75	-5.75	5.75
5	80.5	120.25	39.75	39.75
6	110	127.25	17.25	17.25
7	127	126.50	-0.5	0.5
Total	808.25	853.75	45.5	106.5

Tabelle 15 - Aufwand Übersicht

Die nachfolgende Abbildung zeigt die gleichen Daten in einem Diagramm.

Die blaue Linie stellt die geplanten Stunden dar. In diesen geplanten Stunden ist Timeboxing eingerechnet. Timeboxen bezeichnet das Verschieben von Features, welche für den aktuellen Sprint geplant waren, aber nicht fertig entwickelt werden konnten, in den nächsten Sprint. Wird Timeboxing gemacht, so nimmt der geschätzte Aufwand für den Sprint, aus dem das Ticket verschoben wird, ab.

Die rote Linie zeigt den tatsächlich investierten Aufwand.

Die grüne Linie stellt die Differenz zwischen Geplant und Aufwand dar, die violette Linie zeigt die absolute Abweichung.

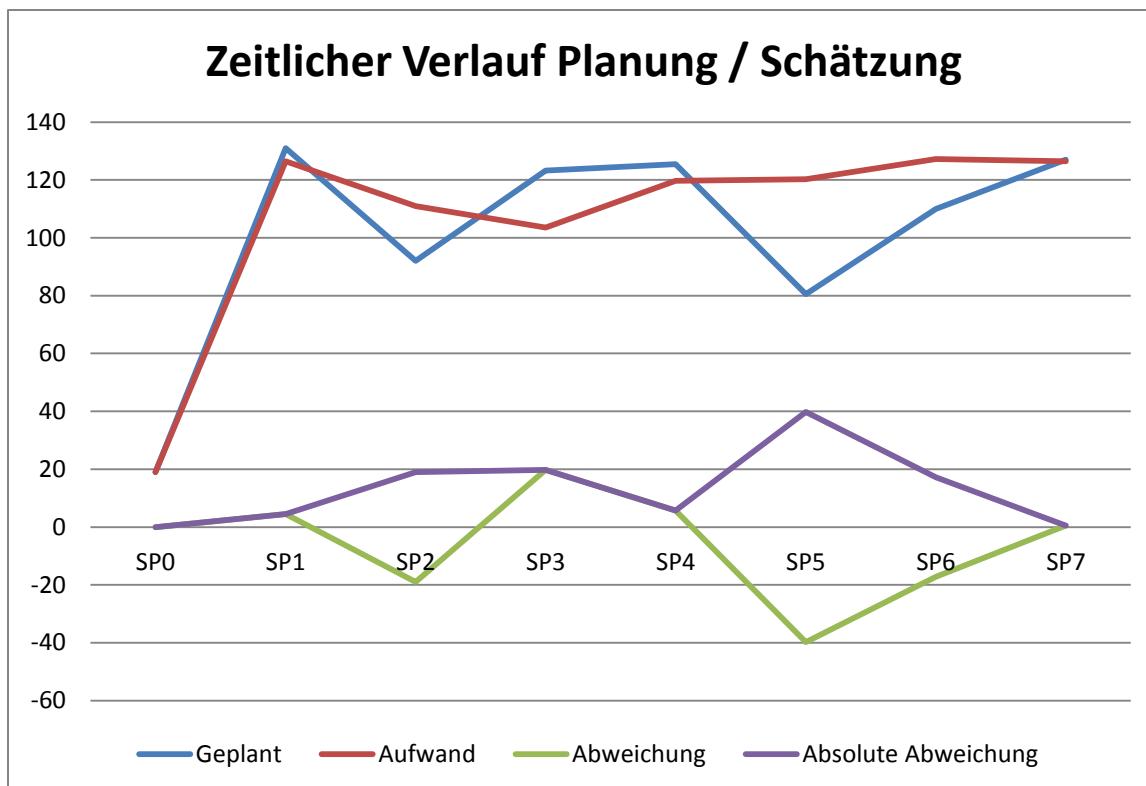


Abbildung 90 - Zeitlicher Verlauf Planung / Schätzung

Nachfolgend werden die einzelnen Sprints detailliert betrachtet und analysiert.

VI.4.1.1 Sprint 0

Der Sprint 0 wurde, wie zu Beginn dieses Unterkapitels bereits erwähnt, bereits vor Semesterbeginn durchgeführt. Er diente dem Kennenlernen der Partnerfirma und der Einarbeitung in die neuen Technologien.

Die folgenden Tickets wurden im Sprint 0 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
460	SPO	Kickoff Meeting	7.5	7.5	31.08.2011	31.08.2011
455	SPO	Planung Sprint	2.5	2.5	31.08.2011	31.08.2011
444	SPO	WPF Einführung	9	9	07.09.2011	07.09.2011
Total			19	19		

Tabelle 16 - Tickets SP 0

VI.4.1.2 Sprint 1

Im ersten Sprint war der Fokus auf den Architekturprototypen gesetzt. Durch die Erarbeitung der Architektur konnten viele Risiken abgedeckt und Unsicherheiten geklärt werden. Da die Implementation dieses Prototyps ohne grössere Probleme verlief, stimmte die Schätzung auch gut mit dem tatsächlich investierten Aufwand überein.

Die folgenden Tickets wurden im Sprint 1 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
456	SP1	Planung Sprint 1	6.5	6.5	19.09.2011	19.09.2011
445	SP1	Meeting, 19.09.11	6	6	19.09.2011	19.09.2011
465	SP1	Entwicklungsumgebung installieren/einrichten	6	6	19.09.2011	20.09.2011
464	SP1	Dokumenterstellung	4	6	19.09.2011	23.09.2011
449	SP1	PDF zu XPS konvertieren	8	7	23.09.2011	23.09.2011
494	SP1	Brainstorming, Interview analysieren	7.5	7.5	26.09.2011	26.09.2011
480	SP1	Interviews durchführen	9	3	26.09.2011	26.09.2011
463	SP1	Meeting, 26.09.11	3	6	26.09.2011	26.09.2011
519	SP1	Planung Sprint	10	10	30.09.2011	30.09.2011
496	SP1	Einarbeitung Scrum	8	6	26.09.2011	30.09.2011
491	SP1	Sitzungsprotokolle	1	1	26.09.2011	30.09.2011
482	SP1	XPS auf Surface darstellen	3	7	21.09.2011	30.09.2011
481	SP1	Interviews erstellen & organisieren	2	1	21.09.2011	30.09.2011
453	SP1	Szenarien	1	1	23.09.2011	30.09.2011
452	SP1	Personas	2	2.5	23.09.2011	30.09.2011
492	SP1	Studium Technologie	4	4.5	19.09.2011	03.10.2011
490	SP1	Flip 1.0 analysieren und dokumentieren	4	3.25	26.09.2011	03.10.2011
489	SP1	Projektmanagement	4	6.5	19.09.2011	03.10.2011
484	SP1	Risiko Management	5	1	19.09.2011	03.10.2011
479	SP1	Administrative Arbeiten	2	2	19.09.2011	03.10.2011
466	SP1	Meeting, 03.10.11	12	13.5	03.10.2011	03.10.2011
454	SP1	Architekturprototyp	20	15.25	19.09.2011	03.10.2011
450	SP1	Vision	3	4	19.09.2011	03.10.2011
Total			131	126.5		

Tabelle 17 - Tickets SP 1

VI.4.1.3 Sprint 2

Im Sprint 2 lag der Schwerpunkt auf dem Creative Workshop zur Erarbeitung eines Papierprototyps und der Nutzer- und Anforderungsanalyse. Der Grund für die grosse Abweichung des effektiven Aufwands gegenüber der Schätzung ist, dass mehr Versuche als ursprünglich geplant zur Erarbeitung des optimalen Interaktionsmodels benötigt wurden.

Die folgenden Tickets wurden im Sprint 2 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
509	SP2	Creative Workshop durchgeführt	12	12	03.10.2011	07.10.2011
446	SP2	Meeting, 10.10.11	6	6	10.10.2011	10.10.2011
570	SP2	Sprint 7 vorgeplant	2	2	03.10.2011	17.10.2011
563	SP2	Guidelines gelesen / überflogen / dokumentiert	2	2	03.10.2011	17.10.2011
543	SP2	Creative Workshop	6	15	03.10.2011	17.10.2011

dokumentiert						
514	SP2	User Stories priorisiert	9	4.5	03.10.2011	17.10.2011
513	SP2	Scrum Poker durchgeführt	3	3	03.10.2011	17.10.2011
512	SP2	User Stories definiert	6	6	03.10.2011	17.10.2011
511	SP2	Papierprototyp getestet und ausgewertet	8	20	03.10.2011	17.10.2011
510	SP2	Papierprototyp erstellt	8	15.5	03.10.2011	17.10.2011
508	SP2	Szenarien erstellt	12	6.5	03.10.2011	17.10.2011
507	SP2	Personas erstellt	3	2	03.10.2011	17.10.2011
506	SP2	Vision erstellt	2	2	03.10.2011	17.10.2011
485	SP2	Risikomanagement nachgeführt	1	2.5	03.10.2011	17.10.2011
467	SP2	Meeting, 17.10.11	6	7.5	17.10.2011	17.10.2011
457	SP2	Sprint 3 geplant	6	4.5	14.10.2011	17.10.2011
Total			92	111.00		

Tabelle 18 - Tickets SP 2

VI.4.1.4 Sprint 3

Der zentrale Punkt des dritten Sprints waren die Grundanforderungen an die Applikation, der Kern der Architektur und die Grundfunktionalität wurden implementiert. Es wurde weniger Zeit benötigt als geplant war.

Die folgenden Tickets wurden im Sprint 3 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
571	SP3	Projektplan und User Stories angepasst	4	6	17.10.2011	23.10.2011
566	SP3	Guidelines gelesen / überflogen / dokumentiert	4	8	18.10.2011	24.10.2011
468	SP3	Meeting, 24.10.11	3	3.25	24.10.2011	24.10.2011
592	SP3	NF Anforderungen dokumentiert	4	2.5	17.10.2011	31.10.2011
583	SP3	SVN Tag SP3 erstellen	0.25	0.25	31.10.2011	31.10.2011
582	SP3	Navigation "Detailansicht → Detailansicht"	8	5.75	17.10.2011	31.10.2011
580	SP3	PN hineinladen stabiler / fehlertoleranter gestaltet	8	7.25	24.10.2011	31.10.2011
579	SP3	UnitTests für Prototypen geschrieben	8	9.5	17.10.2011	31.10.2011
578	SP3	Darstellung der verkleinerten PN mit Bild	16	0	17.10.2011	31.10.2011
577	SP3	Navigation „Detail → Übersicht“	8	6.25	17.10.2011	31.10.2011
576	SP3	Navigation „Übersicht → Detail“	8	6.5	17.10.2011	31.10.2011
575	SP3	Detailansicht PN	16	10.5	17.10.2011	31.10.2011
574	SP3	Übersicht für PN	24	0	17.10.2011	31.10.2011
573	SP3	Anzeige der PNs in Übersicht	8	7.75	17.10.2011	31.10.2011
572	SP3	Bilder aus XPS extrahiert	16	8.5	17.10.2011	31.10.2011
569	SP3	Diskussion Accessability dokumentiert	2	1.25	17.10.2011	31.10.2011

568	SP3	Design Constraints dokumentiert	3	0.75	17.10.2011	31.10.2011
556	SP3	Tests dokumentiert	6	3	17.10.2011	31.10.2011
515	SP3	Domain Model erstellt	4	4.75	17.10.2011	31.10.2011
486	SP3	Risikomanagement nachgeführt	1	0.75	17.10.2011	31.10.2011
458	SP3	Sprint 4 geplant	6	3.5	31.10.2011	31.10.2011
447	SP3	Meeting, 31.10.11	6	7.5	31.10.2011	31.10.2011
Total			123.25	103.5		

Tabelle 19 - Tickets SP 3

VI.4.1.5 Sprint 4

Im Sprint 4 wurden weitere Grundanforderungen umgesetzt. Zusätzlich wurden die Metadaten aggregiert und die Mapping Funktionalität implementiert.

Die folgenden Tickets wurden im Sprint 4 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
594	SP4	Padding / Margin in Übersicht-Buttons entfernen	3	3	04.11.2011	04.11.2011
584	SP4	Meeting, 04.11.11	6	9	04.11.2011	04.11.2011
567	SP4	NF Anforderungen dokumentiert	1.5	2	01.11.2011	04.11.2011
469	SP4	Meeting, 07.11.11	3	3.25	07.11.2011	07.11.2011
597	SP4	Navigation "Detailansicht → Detailansicht"	2	3.5	07.11.2011	08.11.2011
470	SP4	Meeting, 11.11.11	6	6.5	11.11.2011	11.11.2011
459	SP4	Sprint 5 geplant	6	0.5	11.11.2011	11.11.2011
596	SP4	Layout definiert	16	22.25	06.11.2011	14.11.2011
593	SP4	Refactoring	6	8.5	31.10.2011	14.11.2011
591	SP4	Aggregierte Tags anpassbar	8	7.5	31.10.2011	14.11.2011
590	SP4	Tags aggregieren	8	10	31.10.2011	14.11.2011
589	SP4	Filter in Übersicht entfernen	8	4.25	31.10.2011	14.11.2011
588	SP4	Filterkriterium auswählen	16	11.5	31.10.2011	14.11.2011
587	SP4	Filter in Übersicht setzen	8	10.5	31.10.2011	14.11.2011
586	SP4	Tags zu PN angezeigt	16	8	31.10.2011	14.11.2011
585	SP4	PivotViewer untersucht und dokumentiert	2	3.5	31.10.2011	14.11.2011
557	SP4	Tests dokumentiert	3	2.5	31.10.2011	14.11.2011
487	SP4	Risikomanagement nachgeführt	1	0.5	31.10.2011	14.11.2011
448	SP4	Meeting, 14.11.11	6	3	14.11.2011	14.11.2011
Total			125.5	119.75		

Tabelle 20 - Tickets SP 4

VI.4.1.6 Sprint 5

Im fünften Sprint traten Probleme mit den Animationen auf. Diese waren einiges komplizierter als erwartet und beanspruchten dementsprechend mehr Zeit.

Die folgenden Tickets wurden im Sprint 5 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
604	SP5	Scrollerkennung in Übersicht dargestellt	4	29	14.11.2011	21.11.2011
600	SP5	Meeting, 25.11.11	4.5	9	25.11.2011	25.11.2011
462	SP5	Sprint 6 geplant	6	1	25.11.2011	25.11.2011
618	SP5	Anforderungen und Domain Modell Dokumente korrigiert	3	3.5	28.11.2011	28.11.2011
608	SP5	Schöne Darstellung des Filters	8	10.5	14.11.2011	28.11.2011
607	SP5	Animation für Navigation "Detailansicht → Detailansicht"	8	19.75	14.11.2011	28.11.2011
606	SP5	Schöne Darstellung der Tags	8	7.5	14.11.2011	28.11.2011
605	SP5	Lesemodus PN anzeigen	4	12	14.11.2011	28.11.2011
603	SP5	Animierte Navigation „Übersicht → Detail“	16	6.25	14.11.2011	28.11.2011
602	SP5	Filtern nach mehreren Kriterien	8	9	14.11.2011	28.11.2011
601	SP5	Filter in Detailansicht setzen	4	3.5	14.11.2011	28.11.2011
558	SP5	Tests dokumentiert	3	3	14.11.2011	28.11.2011
488	SP5	Risikomanagement nachgeführt	1	1	14.11.2011	28.11.2011
472	SP5	Meeting, 28.11.11	3	5.25	28.11.2011	28.11.2011
Total			80.5	120.25		

Tabelle 21 - Tickets SP 5

VI.4.1.7 Sprint 6

Der Fokus des sechsten Sprints lag auf der Erweiterung der Animationen und dem Implementieren des Lesemodus. Wiederum beanspruchten die Arbeiten mehr Zeit als geplant war. Aus der Auswertung des Usability Tests sind weitere, neue Anforderungen entstanden, zu deren Umsetzung keine Zeit geplant worden war.

Die folgenden Tickets wurden im Sprint 6 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
471	SP6	Meeting, 05.12.11	6	6.25	05.12.2011	05.12.2011
473	SP6	Meeting, 09.12.11	6	5.5	09.12.2011	09.12.2011
637	SP6	SVN Tag SP6 erstellt	0.5	0.5	12.12.2011	12.12.2011
629	SP6	Info View / Easteregg	4	8	08.12.2011	12.12.2011
628	SP6	Codereview durchgeführt	3	3	09.12.2011	12.12.2011
623	SP6	Verbesserungen aus Usability Test gemacht	8	19	05.12.2011	12.12.2011
622	SP6	Dokumente vervollständigt / diverse kleine Anpassungen	4	4	05.12.2011	12.12.2011
621	SP6	Refactoring	5	9	29.11.2011	12.12.2011
620	SP6	Betriebskonzept / Installationsanleitung erarbeitet	4	1.5	28.11.2011	12.12.2011
619	SP6	Funktionale Anforderungen dokumentiert	2	1.25	28.11.2011	12.12.2011
617	SP6	3-D Darstellung der PNs dokumentiert	4	5.5	28.11.2011	12.12.2011

616	SP6	Animation Filter → Übersicht	8	10	28.11.2011	12.12.2011
615	SP6	Anzahl PN bei Übersicht angezeigt	4	5.5	28.11.2011	12.12.2011
613	SP6	Dokument "Realisierung & Test" geschrieben	3	3	28.11.2011	12.12.2011
612	SP6	SE Dok: "Entwurf" geschrieben	6	7.5	28.11.2011	12.12.2011
564	SP6	Usability Tests durchgeführt und protokolliert	12	8.75	02.12.2011	12.12.2011
559	SP6	Tests dokumentiert	3	0.5	28.11.2011	12.12.2011
551	SP6	Aufgabenstellung unterschrieben	1	0.75	28.11.2011	12.12.2011
550	SP6	Lizenzvereinbarung unterschrieben	4	2	28.11.2011	12.12.2011
549	SP6	Erklärung eigenständige Arbeit unterschrieben	1	1	28.11.2011	12.12.2011
546	SP6	Abstract geschrieben	9.5	9.75	12.12.2011	12.12.2011
518	SP6	Risikomanagement nachgeführt	1	0.5	28.11.2011	12.12.2011
517	SP6	Sprint 7 geplant	4	0.5	09.12.2011	12.12.2011
474	SP6	Meeting, 12.12.11	3	4.75	12.12.2011	12.12.2011
614	SP6	Lesemodus PN anzeigen	4	9.25	29.11.2011	13.12.2011
Total			110	127.25		

Tabelle 22 - Tickets SP 6

VI.4.1.8 Sprint 7

Im letzten Sprint galt es, die Dokumentation des Projektes fertig zu stellen.

Die folgenden Tickets wurden im Sprint 7 abgearbeitet:

#	Sprint	Thema	Geplante Stunden	Aufwand in Stunden	Beginn	Abgabedatum
636	SP7	Dokument "Realisierung & Test" geschrieben	3	5	13.12.2011	13.12.2011
635	SP7	Usability Tests durchgeführt und protokolliert	4	3.5	13.12.2011	13.12.2011
640	SP7	Codereview 2 durchgeführt, dokumentiert, implementiert	4	4	16.12.2011	16.12.2011
634	SP7	SE Dok: "Entwurf" geschrieben	16	10	13.12.2011	19.12.2011
633	SP7	Code dokumentiert	5	2	12.12.2011	19.12.2011
632	SP7	Weiterentwicklung dokumentiert	8	5.25	12.12.2011	19.12.2011
475	SP7	Meeting, 19.12.11	3	4.75	19.12.2011	19.12.2011
630	SP7	Abstract geschrieben	3	1.5	20.12.2011	20.12.2011
641	SP7	Allgemeine Korrekturen, kleine Anpassungen	19	16.5	18.12.2011	23.12.2011
639	SP7	Einleitung Technischer Bericht geschrieben	3	2	15.12.2011	23.12.2011
631	SP7	Dokumente zusammenfügen, PDF generieren	10	14.5	12.12.2011	23.12.2011
562	SP7	Video und Wiki Seite erstellt	8	14	12.12.2011	23.12.2011
561	SP7	Tools sind beschrieben	1	1	12.12.2011	23.12.2011
554	SP7	Codereview dokumentiert	4	1	12.12.2011	23.12.2011

552	SP7	Installationsanleitung geschrieben	2	1	12.12.2011	23.12.2011
548	SP7	Extended Management Summary geschrieben	14	10.5	12.12.2011	23.12.2011
547	SP7	Management Summary geschrieben	0	0	12.12.2011	23.12.2011
545	SP7	Persönlicher Bericht geschrieben	12	11.5	12.12.2011	23.12.2011
544	SP7	CD gebrannt und abgegeben	3	3	12.12.2011	23.12.2011
520	SP7	Risikomanagement nachgeführt	1	0.5	12.12.2011	23.12.2011
638	SP7	Codereview Verbesserungen implementiert	4	15	24.12.2011	26.12.2011
Total			127	126.5		

Tabelle 23 - Tickets SP 7

VI.4.2 Personenaufwand

Die Arbeitsaufwände pro Person sind sehr ausgeglichen (siehe Tabelle 24 - Übersicht Personenaufwand und Abbildung 91 - Aufteilung Personenaufwand):

Mitglied	Aufwand in Stunden
Lukas Elmer	295.25
Christina Heidt	275.25
Delia Treichler	283.25
Total	853.75

Tabelle 24 - Übersicht Personenaufwand

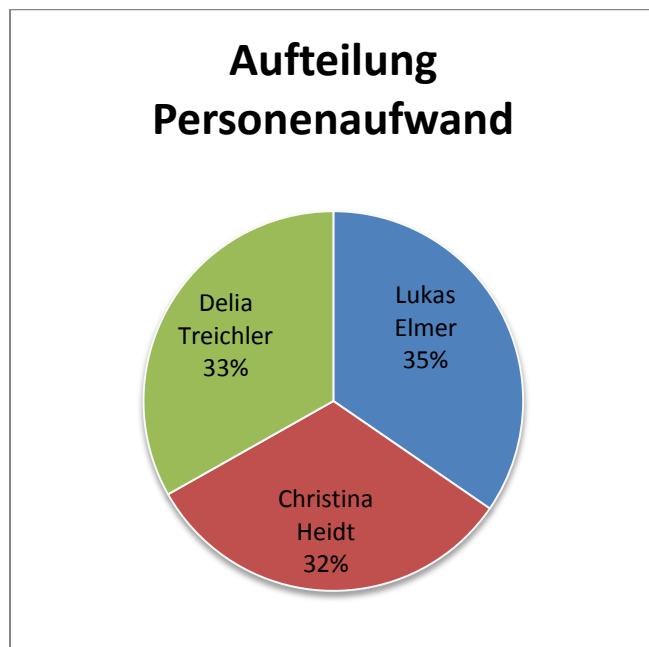


Abbildung 91 - Aufteilung Personenaufwand

Im nachfolgenden Diagramm ist der Verlauf des Aufwands pro Person über alle Sprints ersichtlich.

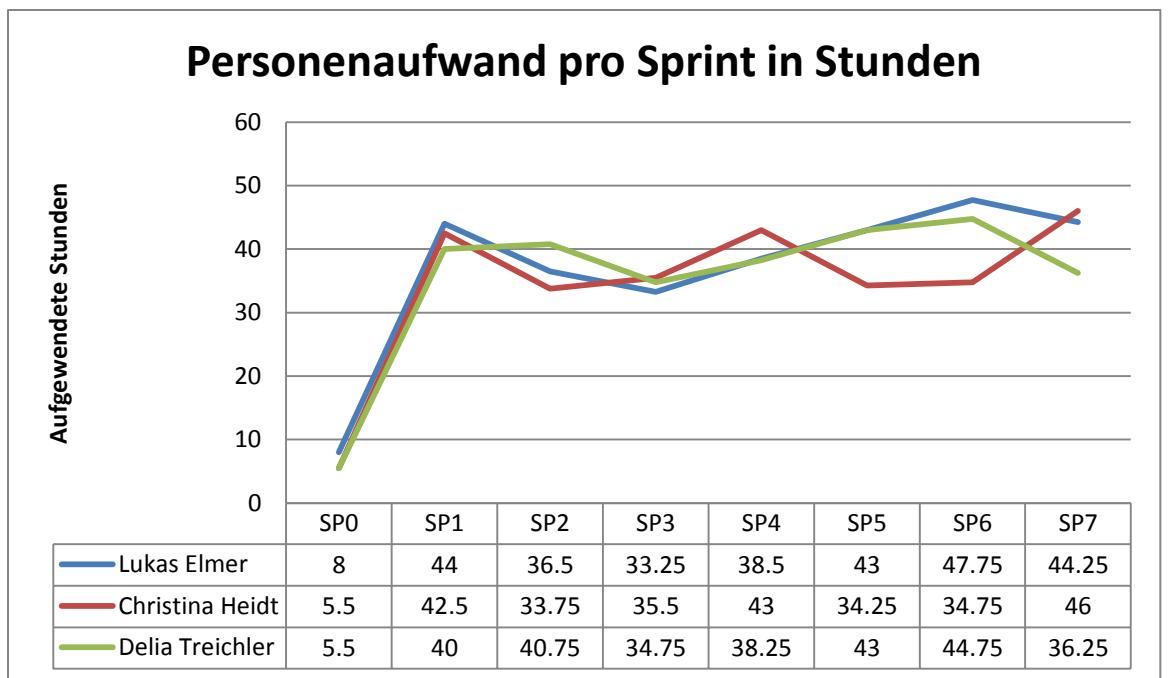


Abbildung 92 - Personenaufwand pro Sprint

VI.4.3 Tätigkeiten

Wie im nachfolgenden Diagramm ersichtlich ist, wurden etwa drei Viertel der Zeit durch die Tätigkeiten Implementation, Dokumentation und Sitzungen aufgewendet. Ebenfalls interessant ist, dass die Sitzungen ganze 14% des Aufwands betragen.

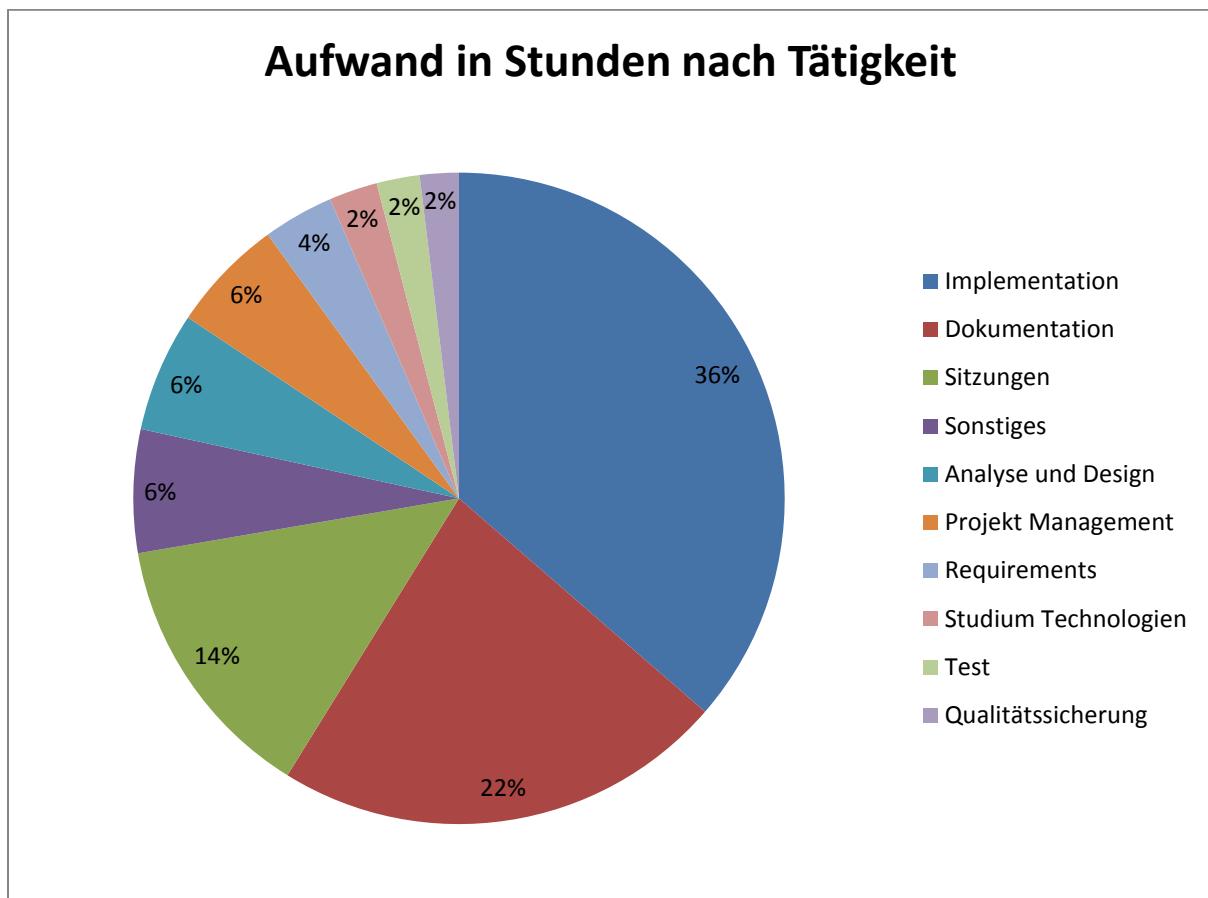


Abbildung 93 - Aufwand in Stunden nach Tätigkeit

Das folgende Diagramm zeigt, dass Lukas Elmer deutlich mehr implementiert hat als die anderen Teammitglieder. Dies liegt vor allem daran, dass er für die Implementation der Perspective Wall und die Korrekturen, die sich aus den Code Reviews ergaben, verantwortlich war.

Christina Heidt fokussierte sich in ihrer Arbeit auf die Erarbeitung der Personas, Szenarien und den Papierprototyp gelegt. Auch die Erstellung des Videos, welcher unter der Aktivität Sonstiges verbucht wurde, gehörte zu ihren Hauptaktivitäten. Zusätzlich war sie für das Protokollieren der Sitzungen verantwortlich, was unter der Aktivität Dokumentation verbucht wurde.

Der Fokus der Arbeit von Delia Treichler lag im Sprint 5 und 6 auf den Animationen, die unter der Aktivität Implementation verbucht wurden. Zusätzlich war sie bei der Projektplanung stets beteiligt. Ansonsten waren ihre Tätigkeiten sehr ausgewogen.

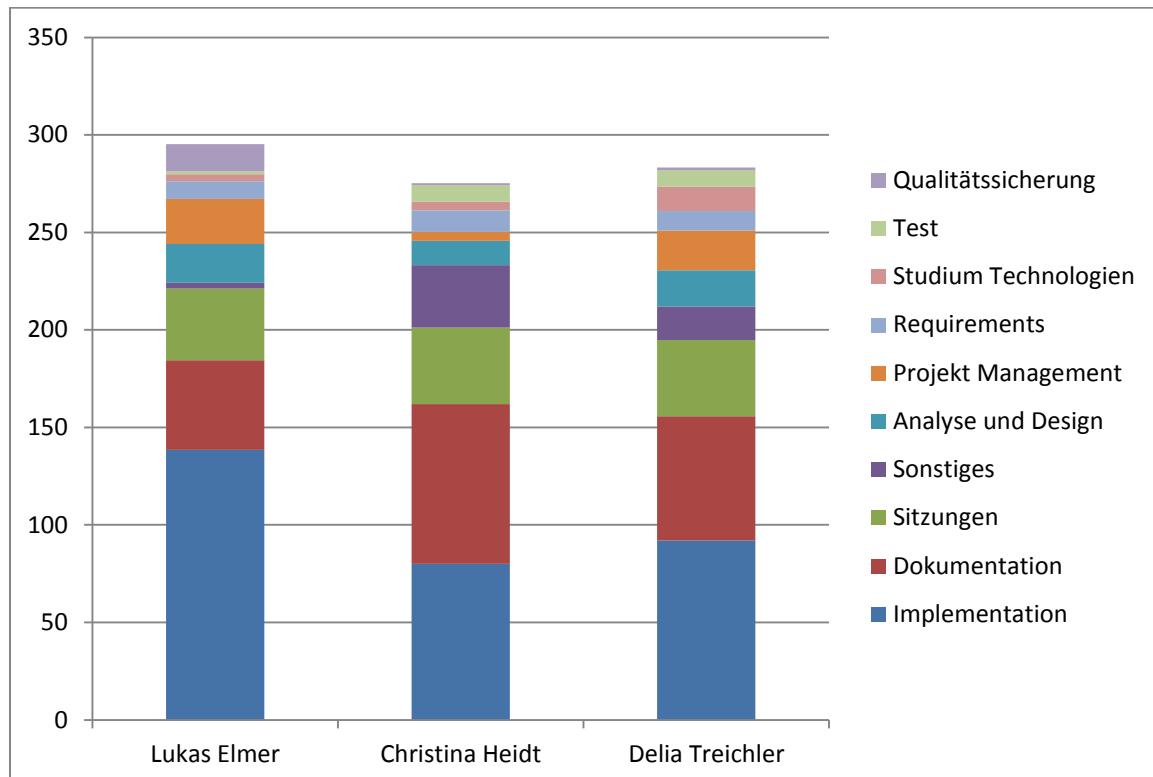


Abbildung 94 - Aktivitäten nach Personen gruppiert

VI.4.4 Arbeitslisten

In diesem Abschnitt werden die Arbeitslisten pro Person mit dem zugehörigen Aufwand aufgelistet.

VI.4.4.1 Lukas Elmer

Version	Ticket	Aufwand in Stunden
SP0		8
	Feature #444: WPF Einführung	3
	Feature #455: Planung Sprint	2.5
	Feature #460: Kickoff Meeting	2.5
SP1		44
	Feature #445: Meeting, 19.09.11	2
	Feature #449: PDF zu XPS konvertieren	7
	Feature #450: Vision	1.5
	Feature #454: Architekturprototyp	10.75

Feature #456: Planung Sprint 1	3.5
Feature #463: Meeting, 26.09.11	2
Feature #464: Dokumenterstellung	0.5
Feature #465: Entwicklungsumgebung installieren/einrichten	2
Feature #466: Meeting, 03.10.11	4.5
Feature #480: Interviews durchführen	1
Feature #482: XPS auf Surface darstellen	0.5
Feature #484: Risiko Management	0.5
Feature #490: Flip 1.0 analysieren und dokumentieren	1.25
Feature #492: Studium Technologie	0.5
Feature #494: Brainstorming, Interview analysieren	2.5
Feature #496: Einarbeitung Scrum	1
Feature #519: Planung Sprint	3
SP2	36.5
Feature #446: Meeting, 10.10.11	2
Feature #457: Sprint 3 geplant	1
Feature #467: Meeting, 17.10.11	2.5
Feature #485: Risikomanagement nachgeführt	2
Feature #506: Vision erstellt	0.5
Feature #507: Personas erstellt	0.5
Feature #508: Szenarien erstellt	0.5
Feature #509: Creative Workshop durchgeführt	4
Feature #510: Papierprototyp erstellt	4
Feature #511: Papierprototyp getestet und ausgewertet	4
Feature #512: User Stories definiert	2
Feature #513: Scrum Poker durchgeführt	1
Feature #514: User Stories priorisiert	3.5
Feature #543: Creative Workshop dokumentiert	6
Feature #563: Guidelines gelesen / überflogen / dokumentiert	1
Feature #570: Sprint 7 vorgeplant	2
SP3	33.25
Feature #447: Meeting, 31.10.11	2
Feature #458: Sprint 4 geplant	1
Feature #468: Meeting, 24.10.11	1
Feature #486: Risikomanagement nachgeführt	0.75
Feature #515: Domain Model erstellt	2
Feature #556: Tests dokumentiert	1
Feature #571: Projektplan und User Stories angepasst	2
Feature #572: Bilder aus XPS extrahiert	8.5
Feature #576: Navigation „Übersicht -> Detail“	1
Feature #577: Navigation „Detail -> Übersicht“	1.5
Feature #579: UnitTests für Prototypen geschrieben	5
Feature #580: PN hineinladen stabiler / fehlertoleranter gestaltet	7.25
Feature #583: SVN Tag SP3 erstellen	0.25
SP4	38.5

Feature #448: Meeting, 14.11.11	1
Feature #469: Meeting, 07.11.11	1
Feature #470: Meeting, 11.11.11	2
Feature #487: Risikomanagement nachgeführt	0.5
Feature #567: NF Anforderungen dokumentiert	0.75
Feature #584: Meeting, 04.11.11	3
Feature #587: Filter in Übersicht setzen	1.5
Feature #590: Tags aggregieren	9.5
Feature #591: Aggregierte Tags anpassbar	7.5
Feature #593: Refactoring	8.5
Feature #594: Padding / Margin in Übersicht-Buttons entfernen	3
Feature #596: Layout definiert	0.25
SP5	43
Feature #472: Meeting, 28.11.11	1.75
Feature #600: Meeting, 25.11.11	3
Feature #604: Scrollerkennung in Übersicht dargestellt	19
Feature #605: Lesemodus PN anzeigen	2
Feature #607: Animation für Navigation "Detailansicht -> Detailansicht"	17.25
SP6	47.75
Feature #471: Meeting, 05.12.11	2
Feature #473: Meeting, 09.12.11	2.5
Feature #474: Meeting, 12.12.11	1.5
Feature #517: Sprint 7 geplant	0.5
Feature #518: Risikomanagement nachgeführt	0.5
Feature #559: Tests dokumentiert	0.5
Feature #564: Usability Tests durchgeführt und protokolliert	1
Feature #612: SE Dok: "Entwurf" geschrieben	7.5
Feature #614: Lesemodus PN anzeigen	5
Feature #617: 3-D Darstellung der PNs dokumentiert	4
Feature #619: Funktionale Anforderungen dokumentiert	0.75
Feature #620: Betriebskonzept / Installationsanleitung erarbeitet	0.5
Feature #621: Refactoring	9
Feature #623: Verbesserungen aus Usability Test gemacht	4
Feature #628: Codereview durchgeführt	1
Feature #629: Info View / Eastereggs	7
Feature #637: SVN Tag SP6 erstellt	0.5
SP7	44.25
Feature #475: Meeting, 19.12.11	1.5
Feature #544: CD gebrannt und abgegeben	1
Feature #545: Persönlicher Bericht geschrieben	4
Feature #554: Codereview dokumentiert	1
Feature #561: Tools sind beschrieben	0.5
Feature #631: Dokumente zusammenfügen, PDF generieren	4.5
Feature #632: Weiterentwicklung dokumentiert	0.25
Feature #634: SE Dok: "Entwurf" geschrieben	3.5

Feature #636: Dokument "Realisierung & Test" geschrieben	3
Feature #638: Codereview Verbesserungen implementiert	15
Feature #640: Codereview 2 durchgeführt, dokumentiert, implementiert	4
Feature #641: Allgemeine Korrekturen, kleine Anpassungen	6

Tabelle 25 - Arbeitsliste Lukas Elmer

VI.4.4.2 Christina Heidt

Version	Ticket	Aufwand in Stunden
SP0		5.5
	Feature #444: WPF Einführung	3
	Feature #460: Kickoff Meeting	2.5
SP1		42.5
	Feature #445: Meeting, 19.09.11	2
	Feature #450: Vision	2
	Feature #452: Personas	2.5
	Feature #453: Szenarien	1
	Feature #454: Architekturprototyp	4.5
	Feature #456: Planung Sprint 1	1.5
	Feature #463: Meeting, 26.09.11	2
	Feature #464: Dokumenterstellung	4.5
	Feature #465: Entwicklungsumgebung installieren/einrichten	2
	Feature #466: Meeting, 03.10.11	4
	Feature #480: Interviews durchführen	1
	Feature #481: Interviews erstellen & organisieren	0.5
	Feature #482: XPS auf Surface darstellen	6.5
	Feature #492: Studium Technologie	1.5
	Feature #494: Brainstorming, Interview analysieren	2.5
	Feature #496: Einarbeitung Scrum	1.5
	Feature #519: Planung Sprint	3
SP2		33.75
	Feature #446: Meeting, 10.10.11	2
	Feature #457: Sprint 3 geplant	0.5
	Feature #467: Meeting, 17.10.11	2.5
	Feature #507: Personas erstellt	1
	Feature #508: Szenarien erstellt	3.75
	Feature #509: Creative Workshop durchgeführt	4
	Feature #510: Papierprototyp erstellt	3.5
	Feature #511: Papierprototyp getestet und ausgewertet	7
	Feature #512: User Stories definiert	2
	Feature #513: Scrum Poker durchgeführt	1
	Feature #514: User Stories priorisiert	0.5
	Feature #543: Creative Workshop dokumentiert	6
SP3		35.5
	Feature #447: Meeting, 31.10.11	2
	Feature #458: Sprint 4 geplant	1
	Feature #468: Meeting, 24.10.11	1
	Feature #515: Domain Model erstellt	2
	Feature #556: Tests dokumentiert	2
	Feature #566: Guidelines gelesen / überflogen / dokumentiert	5.5

Feature #568: Design Constraints dokumentiert	0.5
Feature #569: Diskussion Accessability dokumentiert	1
Feature #571: Projektplan und User Stories angepasst	0.75
Feature #573: Anzeige der PNs in Übersicht	7.75
Feature #576: Navigation „Übersicht -> Detail“	5.5
Feature #579: UnitTests für Prototypen geschrieben	4.5
Feature #592: NF Anforderungen dokumentiert	2
SP4	43
Feature #448: Meeting, 14.11.11	1
Feature #469: Meeting, 07.11.11	1
Feature #470: Meeting, 11.11.11	2
Feature #557: Tests dokumentiert	1.5
Feature #584: Meeting, 04.11.11	3
Feature #585: PivotViewer untersucht und dokumentiert	2
Feature #586: Tags zu PN angezeigt	8
Feature #587: Filter in Übersicht setzen	2
Feature #588: Filterkriterium auswählen	2.5
Feature #589: Filter in Übersicht entfernen	0.5
Feature #596: Layout definiert	19.5
SP5	34.25
Feature #472: Meeting, 28.11.11	1.75
Feature #600: Meeting, 25.11.11	3
Feature #602: Filtern nach mehreren Kriterien	7
Feature #604: Scrollerkennung in Übersicht dargestellt	10
Feature #606: Schöne Darstellung der Tags	1.5
Feature #608: Schöne Darstellung des Filters	8
Feature #618: Anforderungen und Domain Modell Dokumente korrigiert	3
SP6	34.75
Feature #471: Meeting, 05.12.11	2
Feature #473: Meeting, 09.12.11	1.5
Feature #474: Meeting, 12.12.11	1.5
Feature #546: Abstract geschrieben	9
Feature #551: Aufgabenstellung unterschrieben	0.5
Feature #564: Usability Tests durchgeführt und protokolliert	3.5
Feature #615: Anzahl PN bei Übersicht angezeigt	4.5
Feature #616: Animation Filter -> Übersicht	4
Feature #622: Dokumente vervollständigt / diverse kleine Anpassungen	1
Feature #623: Verbesserungen aus Usability Test gemacht	5.25
Feature #628: Codereview durchgeführt	1
Feature #629: Info View / Eastereggs	1
SP7	46
Feature #475: Meeting, 19.12.11	1.5
Feature #520: Risikomanagement nachgeführt	0.5
Feature #544: CD gebrannt und abgegeben	1
Feature #545: Persönlicher Bericht geschrieben	4

Feature #548: Extended Management Summary geschrieben	7.5
Feature #552: Installationsanleitung geschrieben	1
Feature #562: Video und Wiki Seite erstellt	11.5
Feature #630: Abstract geschrieben	0.5
Feature #631: Dokumente zusammenfügen, PDF generieren	4
Feature #632: Weiterentwicklung dokumentiert	2
Feature #633: Code dokumentiert	2
Feature #634: SE Dok: "Entwurf" geschrieben	2
Feature #635: Usability Tests durchgeführt und protokolliert	2.5
Feature #636: Dokument "Realisierung & Test" geschrieben	1
Feature #639: Einleitung Technischer Bericht geschrieben	0.5
Feature #641: Allgemeine Korrekturen, kleine Anpassungen	4.5

Tabelle 26 - Arbeitsliste Christina Heidt

VI.4.4.3 Delia Treichler

Version	Ticket	Aufwand in Stunden
SP0		5.5
	Feature #444: WPF Einführung	3
	Feature #460: Kickoff Meeting	2.5
SP1		40
	Feature #445: Meeting, 19.09.11	2
	Feature #450: Vision	0.5
	Feature #456: Planung Sprint 1	1.5
	Feature #463: Meeting, 26.09.11	2
	Feature #464: Dokumenterstellung	1
	Feature #465: Entwicklungsumgebung installieren/einrichten	2
	Feature #466: Meeting, 03.10.11	5
	Feature #479: Administrative Arbeiten	2
	Feature #480: Interviews durchführen	1
	Feature #481: Interviews erstellen & organisieren	0.5
	Feature #484: Risiko Management	0.5
	Feature #489: Projektmanagement	6.5
	Feature #490: Flip 1.0 analysieren und dokumentieren	2
	Feature #491: Sitzungsprotokolle	1
	Feature #492: Studium Technologie	2.5
	Feature #494: Brainstorming, Interview analysieren	2.5
	Feature #496: Einarbeitung Scrum	3.5
	Feature #519: Planung Sprint	4
SP2		40.75
	Feature #446: Meeting, 10.10.11	2
	Feature #457: Sprint 3 geplant	3
	Feature #467: Meeting, 17.10.11	2.5
	Feature #485: Risikomanagement nachgeführt	0.5
	Feature #506: Vision erstellt	1.5
	Feature #507: Personas erstellt	0.5
	Feature #508: Szenarien erstellt	2.25
	Feature #509: Creative Workshop durchgeführt	4
	Feature #510: Papierprototyp erstellt	8
	Feature #511: Papierprototyp getestet und ausgewertet	9
	Feature #512: User Stories definiert	2
	Feature #513: Scrum Poker durchgeführt	1
	Feature #514: User Stories priorisiert	0.5
	Feature #543: Creative Workshop dokumentiert	3
	Feature #563: Guidelines gelesen / überflogen / dokumentiert	1
SP3		34.75
	Feature #447: Meeting, 31.10.11	3.5
	Feature #458: Sprint 4 geplant	1.5

Feature #468: Meeting, 24.10.11	1.25
Feature #515: Domain Model erstellt	0.75
Feature #566: Guidelines gelesen / überflogen / dokumentiert	2.5
Feature #568: Design Constraints dokumentiert	0.25
Feature #569: Diskussion Accessability dokumentiert	0.25
Feature #571: Projektplan und User Stories angepasst	3.25
Feature #575: Detailansicht PN	10.5
Feature #577: Navigation „Detail -> Übersicht“	4.75
Feature #582: Navigation "Detailansicht -> Detailansicht"	5.75
Feature #592: NF Anforderungen dokumentiert	0.5
SP4	38.25
Feature #448: Meeting, 14.11.11	1
Feature #459: Sprint 5 geplant	0.5
Feature #469: Meeting, 07.11.11	1.25
Feature #470: Meeting, 11.11.11	2.5
Feature #557: Tests dokumentiert	1
Feature #567: NF Anforderungen dokumentiert	1.25
Feature #584: Meeting, 04.11.11	3
Feature #585: PivotViewer untersucht und dokumentiert	1.5
Feature #587: Filter in Übersicht setzen	7
Feature #588: Filterkriterium auswählen	9
Feature #589: Filter in Übersicht entfernen	3.75
Feature #590: Tags aggregieren	0.5
Feature #596: Layout definiert	2.5
Feature #597: Navigation "Detailansicht -> Detailansicht"	3.5
SP5	43
Feature #462: Sprint 6 geplant	1
Feature #472: Meeting, 28.11.11	1.75
Feature #488: Risikomanagement nachgeführt	1
Feature #558: Tests dokumentiert	3
Feature #600: Meeting, 25.11.11	3
Feature #601: Filter in Detailansicht setzen	3.5
Feature #602: Filtern nach mehreren Kriterien	2
Feature #603: Animierte Navigation „Detail -> Übersicht“	6.25
Feature #605: Lesemodus PN anzeigen	10
Feature #606: Schöne Darstellung der Tags	6
Feature #607: Animation für Navigation "Detailansicht -> Detailansicht"	2.5
Feature #608: Schöne Darstellung des Filters	2.5
Feature #618: Anforderungen und Domain Modell Dokumente korrigiert	0.5
SP6	44.75
Feature #471: Meeting, 05.12.11	2.25
Feature #473: Meeting, 09.12.11	1.5
Feature #474: Meeting, 12.12.11	1.75
Feature #546: Abstract geschrieben	0.75
Feature #549: Erklärung eigenständige Arbeit unterschrieben	1

Feature #550: Lizenzvereinbarung unterschrieben	2
Feature #551: Aufgabenstellung unterschrieben	0.25
Feature #564: Usability Tests durchgeführt und protokolliert	4.25
Feature #613: Dokument "Realisierung & Test" geschrieben	3
Feature #614: Lesemodus PN anzeigen	4.25
Feature #615: Anzahl PN bei Übersicht angezeigt	1
Feature #616: Animation Filter -> Übersicht	6
Feature #617: 3-D Darstellung der PNs dokumentiert	1.5
Feature #619: Funktionale Anforderungen dokumentiert	0.5
Feature #620: Betriebskonzept / Installationsanleitung erarbeitet	1
Feature #622: Dokumente vervollständigt / diverse kleine Anpassungen	3
Feature #623: Verbesserungen aus Usability Test gemacht	9.75
Feature #628: Codereview durchgeführt	1
SP7	36.25
Feature #475: Meeting, 19.12.11	1.75
Feature #544: CD gebrannt und abgegeben	1
Feature #545: Persönlicher Bericht geschrieben	3.5
Feature #548: Extended Management Summary geschrieben	3
Feature #561: Tools sind beschrieben	0.5
Feature #562: Video und Wiki Seite erstellt	2.5
Feature #630: Abstract geschrieben	1
Feature #631: Dokumente zusammenfügen, PDF generieren	6
Feature #632: Weiterentwicklung dokumentiert	3
Feature #634: SE Dok: "Entwurf" geschrieben	4.5
Feature #635: Usability Tests durchgeführt und protokolliert	1
Feature #636: Dokument "Realisierung & Test" geschrieben	1
Feature #639: Einleitung Technischer Bericht geschrieben	1.5
Feature #641: Allgemeine Korrekturen, kleine Anpassungen	6

Tabelle 27 - Arbeitsliste Delia Treichler

Project Flip 20

Studienarbeit

VII. Anhang