

Design of an Optimized Radiotherapy Overview for Usability and Performance

Bachelorarbeit
Frühjahrssemester 2014



Slobodan Stojkovic
Stefan Leimgruber



Betreuer
Projektpartner
Experte
Gegenleser
Abteilung
Themengebiet

Prof. Dr. Luc Bläser
Varian Medical Systems, Stephan Huber
Jean-Daniel Merkli
Prof. Hansjörg Huser
Informatik
Software

Inhaltsverzeichnis

Abstract	2
Aufgabenstellung	3
Erklärung der eigenständigen Arbeit	7
Vereinbarung Urheber- & Nutzungsrechte	8
Management Summary	9
Projektmanagement	10
Team	10
Projektplan	11
Arbeitspakete	13
Risiken	14
Continuous Integration	16
Analyse	17
Domänenanalyse	17
Anforderungsanalyse	19
Feldforschung	20
Lösungsvarianten	22
Gegenüberstellung	34
Entscheidung/Kontroverse	35
Architektur	36
Logische Architektur	36
GUI-Architektur	38
Deklaration Fremd-Code	40
Datenmodell	41
Testing	44
Resultat	46
Auswertung Analysephase	46
Vergleich Ausgangslage / Resultat	47
Umgesetzte Features	48
Weitere Ideen / Open Issues	52
Auswertung Arbeitsaufwand	53
Glossar	54
Literaturverzeichnis	55
Abbildungsverzeichnis	56

Abstract

Ausgangslage

Varian Medical Systems ist der weltweit führende Hersteller von medizinischen Linearbeschleunigern und Software zur Krebstherapie. RT Summary ist eine Anwendung, welche sich im gesamten klinischen Informationssystem befindet. Diese Anwendung begleitet den Patienten in vielen Teilschritten seiner Krebsbehandlung. Von der Planungsübersicht und Behandlungszustimmung bis hin zur Rückverfolgung der ausgeführten Behandlungen unterstützt diese Anwendung den medizinischen Physiker, den Therapeuten und den zuständigen Arzt bei der Ausführung ihrer Aufgaben.

Aufgabenstellung

Die Anwendung RT Summary wird zurzeit produktiv in über 1700 Institutionen eingesetzt. Da die Performance und Usability der aktuellen Anwendung nicht mehr den Anforderungen entspricht, besteht unsere Aufgabe grundsätzlich aus zwei Teilen. Der erste Teil befasst sich mit der Erarbeitung eines Usability-Konzeptes, welches neue Varianten aufzeigen soll, wie die bestehenden Daten zeitgemäss und auch performant dargestellt werden können. Im zweiten Teil der Arbeit geht es darum dieses erarbeitete Konzept auch umzusetzen.

Vorgehen

Um die Anzeige effizient und auch adressatengerecht darstellen zu können, analysierten wir in einem ersten Schritt die Testdaten und das Domainmodell des Projektpartners. Mit agiler Entwicklung von GUI-Prototypen basierend auf den Analyseergebnissen haben wir stets bei unserem Projektpartner die Akzeptanz abgeklärt. Am Schluss entschieden wir uns für eine Variante mit vertikaler Navigation und zusammenklappbaren Datenelementen. Wir fokussierten uns bei der Implementierung bereits auf allfällige Performance-Engpässe, um diese von Beginn an unter Kontrolle zu halten.

Ergebnisse

Die Lösung wurde im Visual Studio 2013 mit WPF/C# und dem .NET Framework 4.5 entwickelt. Sie realisiert die Übersicht der Pläne in vertikaler Zeitachse, dabei können die Pläne aufgeklappt werden um mehr Informationen zu den Behandlungen zu erhalten oder zugeklappt werden um Informationen auszublenden. Bei aufgeklappter Ansicht werden die Behandlungen mit detaillierten Informationen auf Tagesbasis angezeigt. Von da aus besteht die Möglichkeit mehr Informationen zu einem Tag zu erhalten. Auf der Top-Level-Ansicht werden die Zusammenfassungen der Ausnahmen/Änderungen und Bildaufnahmen angezeigt.

Das Resultat ist gelungen: Sämtliche Anforderungen an die neu zu erstellende Anwendung wurden implementiert. Die Anwendung lässt sich wegen einer gemeinsamen Zwischenschnittstelle leicht in das bestehende Informationssystem integrieren und ist mit den vorhandenen Daten kompatibel. Das Design entspricht ebenfalls dem gewünschten „State of the Art“ und passt ästhetisch perfekt in das bestehende Informationssystem.

Aufgabenstellung

Aufgabenstellung Bachelorarbeit für Stefan Leimgruber und Slobodan Stojkovic:

Design of an Optimized Radiotherapy Overview for Usability and Performance

1. Auftraggeber und Betreuer

Diese Bachelorarbeit findet in Zusammenarbeit mit Varian Medical Systems, Imaging Laboratory (iLab), Baden-Dättwil statt.

Ansprechpartner Auftraggeber:

- Stephan Huber, Varian Medical iLab, Team Leader

Betreuer HSR:

- Prof. Dr. Luc Bläser, Institut für Software

2. Ausgangslage

Varian Medical Systems, ist der weltweit führende Hersteller von medizinischen Linearbeschleunigern und Software zur Krebstherapie und weiteren medizinischen Anwendungen wie Radiotherapie, Radiochirurgie und Brachytherapy. Varian ist auch der führende Anbieter von Röntgenröhren für medizinische, wissenschaftliche und industrielle Bildapplikationen und beschäftigt weltweit mehr als 6000 Mitarbeiter..

Eine zentrale Komponente des klinischen Informationssystems ist die Radiotherapie-Behandlungsübersicht, bei dem Ärzte und medizinisches Personal über die kommenden und vergangenen Einsätze wie z.B. medizinische Untersuchungen, diagnostische Bildaufnahmen, Strahlen-Behandlung und Informationen zu Strahlenfraktionen, Maschinen-Setups, Strahlendosen etc. effizient und präzise Überblick erhalten und bei Bedarf weitere Detail-Informationen beziehen können.

Diese Applikation soll nun für die nächste Version neu entworfen und implementiert werden, wobei sich zwei zentrale Anforderungen stellen: Einerseits muss die Anwendung hinsichtlich Performance besonders optimiert sein, zumal diese eine sehr grosse Datenmenge mit einer weiten Suchbreite (Zeit, andere Kriterien) verarbeiten und schnell interaktiv präsentieren soll. Zweitens soll hinsichtlich der Usability neue Ansätze entwickelt werden, um ein möglichst gut verständliches, wohl strukturiertes und einfach bedienbares User Interface anzubieten.

Das neue Informationssystem wird basierend auf .NET, C# und WPF realisiert.

3. Ziele und Aufgabenstellung

Die Aufgabe dieser Arbeit ist es, eine komplett neue optimierte Anwendung für Varian Radiotherapy-Übersicht zu entwickeln. Dabei bestehen vor allem zwei grössere Herausforderungen und Ziele:

- Usability: Es sollen verschiedene bekannte und neuartige Konzepte entwickelt und untersucht werden, inwiefern diese den Benutzer optimal bei der Übersicht und Auskunft über medizinische Ereignisse und Behandlungen unterstützt.
- Performance: Die Vielzahl der Daten und Suchbreite soll möglichst schnell und ohne sichtbare Wartezeiten für den Benutzer präsentiert und von diesem navigiert werden. Dazu sind geeignete Ansätze in der Architektur (Ladestrategien, Caching etc.) zu finden.

Folgende spezifische Ziele werden vorgegeben:

- Analyse der funktionalen und nicht funktionalen Anforderungen sowie der Anwendungsdomäne der neu zu entwickelnden Radiotherapie-Software-Komponente.
- Studie von Konzepten und Modellen hinsichtlich verbesserter Usability. Auswahl der am besten geeigneten Ansätze in Absprache mit den Betreuer und Auftraggeber.
- Analyse der Performance-Charakteristiken und Lösungsansätze für effizientes Laden, Verarbeiten und Präsentieren.
- Design der Software-Architektur für Umsetzung der neuen Radiotherapie-Übersichts-Anwendung mit optimiertem Usability-Ansatz und optimierter Performance.
- Implementierung der neuen Anwendung als Prototyp auf Basis .NET, C# und WPF gemäss erarbeiteten prioritären Anforderungen und mit Umsetzung des gewählten Usability und Performance-Designs.
- Validierung des Systems mit automatisierten Unit und Integration Tests und/oder System-Tests.
- Vorgehen nach modernen Software-Engineering Methoden mit Source Control, Build Server usw.

4. Zur Durchführung

Mit dem HSR-Betreuer finden in der Regel zweiwöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen. Besprechungen mit dem Auftraggeber werden nach Bedarf durchgeführt.

Alle Besprechungen sind von den Studenten mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer und dem Auftraggeber per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsresultate in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsresultate erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

5. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen (siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html?&L=0>). Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Die Dokumentation ist vollständig auf CD/DVD in 3 Exemplaren abzugeben. Auf Wunsch ist für den Auftraggeber eine gedruckte Version zu erstellen.

6. Termine

Siehe auch Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html?&L=0>

17.02.14	Beginn der Bachelorarbeit, Ausgabe der Aufgabenstellung durch die Betreuer.
Mai 2014	Fotoshooting. Genauere Angaben erteilt die Kommunikationsstelle rechtzeitig.
06.06.14	Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei. Die Studierenden erhalten vorgängig vom Studiengangsekretariat die Aufforderung und die Zugangsdaten zur Online-Erfassung des Abstracts für die Broschüre. Die Studierenden senden per Email das A0-Poster zur Prüfung an ihren Examinator/Betreuer. Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.
11.06.14	Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract für die Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.
13.06.14	Abgabe des Berichtes an den Betreuer bis 12.00 Uhr. Fertigstellung des A0-Posters bis 12.00 Uhr.
13.06.14	Präsentation der Bachelorarbeiten, 16 bis 20 Uhr

7. Beurteilung

Eine erfolgreiche Bachelorarbeit zählt 12 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von ca. 25 bis 30 Stunden budgetiert. Für die Modulbeschreibung der Bachelorarbeit siehe auch https://unterricht.hsr.ch/staticWeb/allModules/19419_M_BAI.html.

Für die Beurteilung sind die HSR-Betreuer verantwortlich.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung	1/6
2. Berichte (Abstract, Mgmt Summary, technischer u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation	1/6
3. Inhalt *)	3/6
4. Mündliche Prüfung zur Bachelorarbeit	1/6

*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit mit den Studierenden festgelegt.

Im Übrigen gelten die Bestimmungen der Abt. Informatik zur Durchführung von Studienarbeiten.

Rapperswil, den 12. Februar 2014

Der verantwortliche Dozent


Prof. Dr. Luc Bläser
Institut für Software
Hochschule für Technik Rapperswil

Erklärung der eigenständigen Arbeit

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.
- dass wir keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben.

Rapperswil, 13. Juni 2014



Slobodan Stojkovic



Stefan Leimgruber

Vereinbarung Urheber- & Nutzungsrechte



Vereinbarung über das geistige Eigentum

1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der Bachelorarbeit *Design of an Optimized Radiotherapy Overview for Usability and Performance* unter der Betreuung von Prof. Dr. Luc Bläser mit dem Industriepartner Varian Medical Systems (unten auch Varian genannt) geregelt.

2. Eigentumsrechte am Code

Die Studenten übertragen die Eigentumsrechte des in dieser Arbeit entwickelten Programmcodes der Firma Varian Medical Systems.

2. Nutzungsrechte

Varian kann alle in dieser Arbeit entwickelten Ergebnisse uneingeschränkt und unentgeltlich nutzen, weiterentwickeln und vermarkten. Alle Nutzungs- und Verwertungsrechte an dem entwickelten Programmcode verbleiben ausschliesslich bei Varian.

2. Publikationsrechte

Sowohl die Studenten als auch die HSR dürfen die entwickelten Ideen, Konzepte, Analysen und Beschreibungen – jedoch nicht der Programmcode selbst oder von Varian explizit als vertraulich deklarierte Daten und Informationen – in eigenen Berichten, Poster oder ähnlichem mit Nennung des Industriepartners Varian uneingeschränkt und unentgeltlich publizieren. Publikationen sind vorgängig Varian zu unterbreiten, damit diese prüfen können, dass es keine vertraulichen Daten oder Informationen enthält.

Dättwil, den 19.02.14

Der Student Stefan Leimgruber

Dättwil, den 19.02.14

Der Student Slobodan Stojkovic

Dättwil, den 19.2.14

Der Betreuer der Studienarbeit
Prof. Dr. Luc Bläser

Dättwil, den 19.02.14

Stephan Huber, Varian Medical iLab

Management Summary

Ausgangslage

Varian Medical Systems ist der weltweit führende Hersteller von Geräten für Krebstherapien. Für den gesamten Prozess rund um die Krebstherapie bietet Varian Medical Systems verschiedene Anwendungen an, welche von Personen in unterschiedlichen Berufsgattungen genutzt werden. RT Summary ist eine dieser Anwendungen, welche die Behandlungen visuell darstellt und einen Überblick über vergangene, laufende und zukünftige Behandlungen Auskunft gibt. RT Summary wird zurzeit produktiv in über 1700 Institutionen eingesetzt. Die Benutzerführung ist beim aktuellen Stand etwas rudimentär und hat zum Teil relativ lange Ladezeiten. Das Ziel dieser Arbeit besteht darin, diese Anwendung neu zu designen, so dass sie eine deutlich bessere Benutzerführung mit möglichst hoher Performance ermöglicht.

Vorgehen

Während der Erarbeitung der Daten und Analyse der bestehenden Anwendung wurden Prototypen auf Papier und als klickbare Demo-Applikationen erstellt. Diese wurden in einem wiederholenden Prozess mit dem Betreuer und Projektpartner evaluiert und gegen die Anforderungen geprüft. Dadurch entstanden insgesamt elf unterschiedliche Lösungsvarianten, unter welchen die optimale Variante ausgewählt wurde. Diese stellt die Pläne in einer vertikalen Zeitachse dar und bietet somit eine perfekte Übersicht über alle Pläne, wobei die Position von parallelen Plänen automatisch berechnet wird. Die Pläne enthalten weitere Informationen wie Bildaufnahmen und Angaben zu Ausnahmen oder Änderungen der einzelnen Behandlungen, welche bei Bedarf aufgeklappt werden können, um eine minutengenaue Zurückverfolgung zu ermöglichen. Dieser Ansatz wurde in WPF basierend auf dem .NET Framework implementiert.

Resultate

Zusammenfassend finden wir, dass das erstellte Konzept und die Entwicklung in die richtige Richtung gegangen sind, alle Anforderungen und wichtigen Features umgesetzt worden sind und schliesslich eine optimale Lösung in Bezug auf Benutzerführung erreicht worden ist. Die neue Anwendung lässt sich technisch problemlos in die bestehende Landschaft der Anwendungen integrieren. Die Ladezeit konnte zudem mit unserer Lösung drastisch reduziert werden und stellt dadurch in der Anwendung kein Problem mehr dar.

Projektmanagement

Team



Slobodan Stojkovic

23. Oktober 1989

Skills

- Microsoft .Net C#
- Microsoft SQL
- Software-Engineering

Background

- Wissenschaftliche Hilfskraft, Institut für Software HSR
- Prüfungsexperte Prüfungskommission Kanton Zürich (PK19)
- Informatiker Richtung Applikationsentwicklung



Stefan Leimgruber

31. August 1987

Skills

- Microsoft .Net C#
- WCF
- Software-Engineering

Background

- Siemens Schweiz AG
- Siemens IT Solutions and Services AG
- Informatiker Richtung Systemtechnik

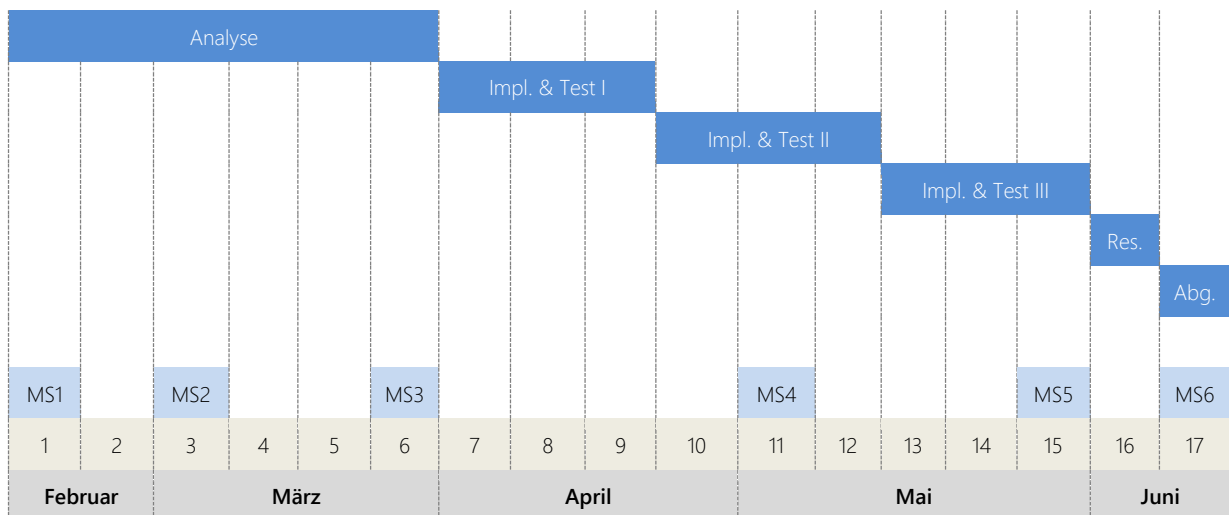
Projektplan

Wir entschieden uns für eine angepasste RUP-Projektmanagementmethode, welche sich bereits bei unserer Studienarbeit bewährt hat. Nach einer Analyse-Phase, in welcher die Stossrichtung festgelegt wird, implementieren wir den evaluierten Ansatz, wobei die Implementation in drei Phasen unterteilt ist. Während der Implementation werden wir ebenfalls den bereits implementierten Code testen. Zur Sicherheit haben wir eine Reserve-Phase eingeplant und die letzte Woche widmen wir ganz der Abgabe.

Zu Beginn des Projektes sind wir folgendem Projektplan gestartet und haben den schliesslich im Verlaufe des Projektes überarbeitet.

Ausgangslage

PHASE	WOCHE	DAUER
Analyse	17.02.2014 – 31.03.2014	6 Wochen
Implementation & Test I	31.03.2014 – 21.04.2014	3 Wochen
Implementation & Test II	21.04.2014 – 12.05.2014	3 Wochen
Implementation & Test III	12.05.2014 – 02.06.2014	3 Wochen
Reserve	02.06.2014 – 09.06.2014	1 Woche
Abgabe	09.06.2014 – 13.06.2014	1 Woche

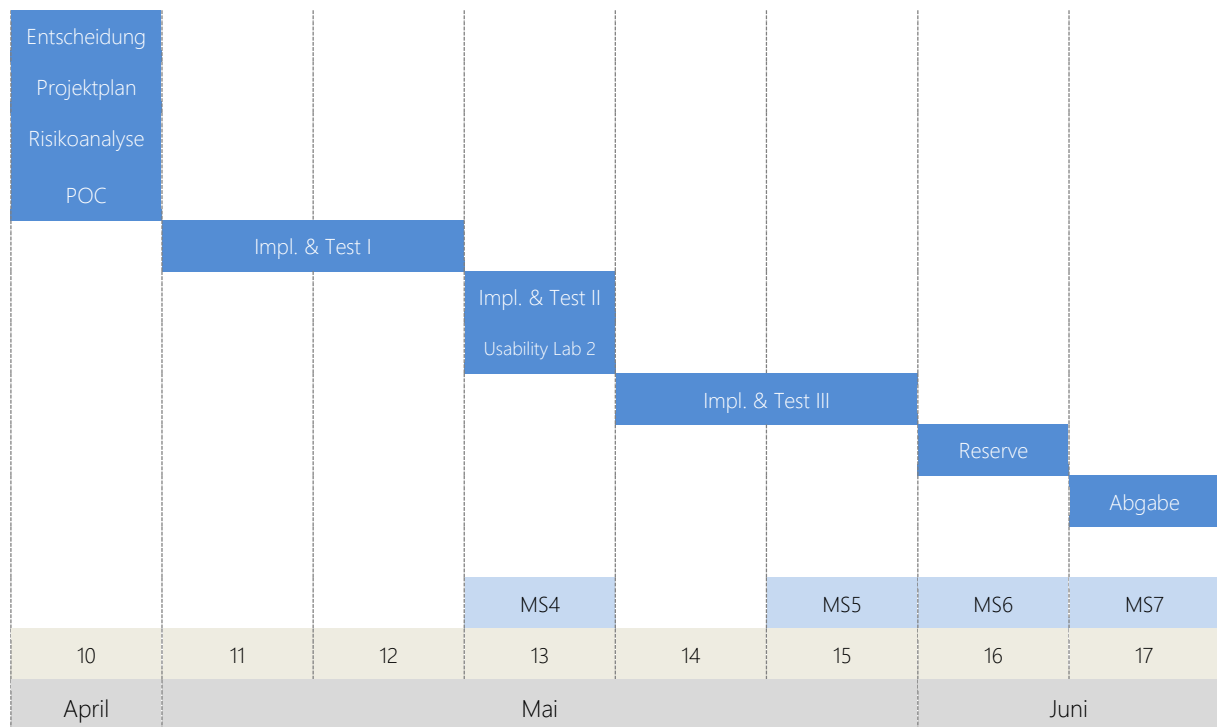


MEILENSTEIN	RESULTATE	TERMIN
MS1 Kickoff	Konkrete Aufgabenstellung	24.02.2014
MS2 Zwischenstand Analyse	Wireframes für Usability-Konzept	10.03.2013
MS3 Ende Analyse	Prototypen für auserwählte Usability-Konzepte	31.03.2013
MS4 Zwischenstand Implementation	Wesentliche Features sind implementiert	02.05.2014
MS5 Ende Implementation	Alle geplanten Features sind implementiert	31.05.2014
MS6 Abschluss Projekt	Abgabe Bericht	13.06.2014

Überarbeitung

Da es in der 9. Woche eine grundlegende Umstellung des zu verfolgenden Ansatzes gab (es dauerte länger als geplant, eine geeignete Lösung zu finden), entschieden wir uns dafür den Projektplan für den Rest der Projektzeit (ab Woche 10) neu zu planen und umzustellen.

PHASE	WOCHE	DAUER
Überarbeitung Projektplan	21.04.2014 – 28.04.2014	1 Woche
Implementation & Test I	28.04.2014 – 12.05.2014	2 Wochen
Implementation & Test II	12.05.2014 – 19.05.2014	1 Woche
Usability Lab 2		
Implementation & Test III	19.05.2014 – 02.06.2014	2 Wochen
Reserve	02.06.2014 – 09.06.2014	1 Woche
Abgabe	09.06.2014 – 13.06.2014	1 Woche





MEILENSTEIN	RESULTATE	TERMIN
MS4 - Erste Version der Implementierung - Usability Lab 2 durchgeführt	Die Idee des Ansatzes ist ersichtlich jedoch ohne Detailschliff. Durchführung des Usability Lab 2.	16.05.2014
MS5 - Überarbeitete Implementierung - Bericht der Bachelorarbeit in erster Fassung	Implementierung soweit bereit für den Code-Review. Bericht als Entwurf zum Review bereit.	30.05.2014
MS6 Entwurf Abstract für Diplombroschüre	Entwurf des Abstracts an Betreuer.	06.06.2014
MS7 Abschluss Projekt	Abgabe Bericht.	13.06.2014









Arbeitspakete

Die Arbeitspakete wurden während der gesamten Arbeit jeweils angepasst und erweitert. Somit ist dieser Stand der Arbeitspakete, der Letzte. (Spalten: O/C = Open ✓ / Closed ✓ / Analyse / Implementation / Sitzungen / Doku / Testen)

NR	BESCHREIBUNG	AUFW. (H)	O/C
1	Analyse Vorlagen (Testdaten, RT Summary, etc..)	40.00	✓
2	Projektmanagement (Zeitplan erstellen, Zeiterfassung vorbereiten, etc.)	20.00	✓
3	Risikomanagement	15.00	✓
4	Ideensammlung/Variantenfindung	30.00	✓
5	Wireframes von potenziellen Ideen	30.00	✓
6	2-3 Wireframes auswählen und weiterverfolgen (genauer spezifizieren)	20.00	✓
7	Prototypen implementieren	40.00	✓
8	SWOT-Analysen der Prototypen	15.00	✓
9	Einarbeiten in Technologien	20.00	✓
10	Usability Labs planen (Aufgaben, Walkthrough, etc..) und durchführen	20.00	✓
11	ViewModels implementieren	40.00	✓
12	(Batches für) Exceptions- und Images-Anzeige	15.00	✓
13	Detailinformationen (Session-Details) planen/implementieren	20.00	✓
14	Performance-Optimierung	5.00	✓
15	Patienteninformationen (Name, Vorname, Foto, Alter, Gewicht, etc..) Darstellung planen und Implementierung	6.00	✓
16	Exceptions-Darstellung planen und implementieren	10.00	✓
17	Planübersicht	10.00	✓
18	Fractions-Ansicht	10.00	✓
19	Plan-Status anzeigen	6.00	✓
20	Course-Unterscheidung in der Anzeige planen/implementieren	8.00	✓
21	Tagesansicht planen/implementieren	15.00	✓
22	Konzept für Zeitachse erstellen/implementieren	6.00	✓
23	Plans ohne Datum anzeigen	9.00	✓
24	Filterfunktion (nach Datum, Fraction, Dosis, etc..)	16.00	✓
25	Suchfunktion (bestimmte Daten, Datum, etc..)	10.00	✓
26	Einstiegspunkt für Doctor und Therapist	10.00	✓
27	Prozess für Image Approval definieren und implementieren	20.00	✓
28	Implementation der Design-Wünsche seitens Varian	30.00	✓
29	Redesign der gesamten Applikation	20.00	✓
30	Code-Anpassungen anhand von Code-Review von Prof. Dr. L. Bläser	10.00	✓
31	Code-Reviews, Refactoring, etc..	30.00	✓
32	Testen der ViewModels (Unit Tests und Manuelle Tests)	60.00	✓
33	Bericht schreiben	60.00	✓
34	Abstract für Diplombroschüre verfassen	8.00	✓
35	Poster erstellen	6.00	✓
36	Abgabe vorbereiten (Drucken, Binden, etc..)	6.00	✓
37	Sitzungen	60.00	✓

Risiken

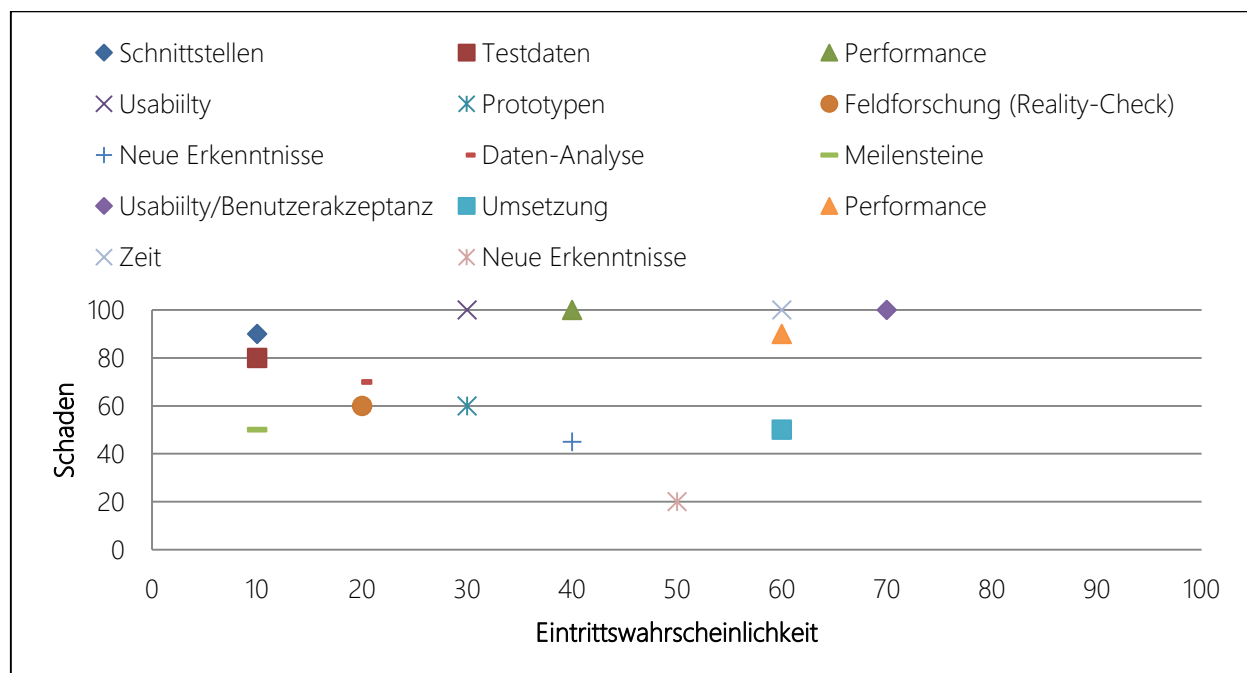
Es war uns wichtig, von Beginn an die Risiken zu kennen und natürlich auch wie diese vorgebeugt werden können. Des Weiteren haben wir den Schaden (Spalte „S“) eingeschätzt. Dadurch entstand die unten aufgeführte Tabelle, welche den Stand der Risiken zu Beginn des Projektes darstellt. (Spalte E = Eingetreten  / Nicht Eingetreten )

RISIKO	BESCHREIBUNG	VORBEUGUNG	S	E
Schnittstellen	Die Schnittstellen zu den Umsystemen (Daten, etc.) werden nicht korrekt geliefert.	Die Schnittstellen werden im Voraus mit dem Projektpartner genau abgeklärt bzw. durch gutes Layering kleiner Aufwand für Änderung.	90	
Testdaten	Die Testdaten entsprechen nicht den realen Daten, bzw. decken nicht die meisten realen Daten ab, somit könnten "Spezialfälle" nicht entdeckt/getestet werden.	Die Testdaten werden so früh wie möglich und auch so umfangreich wie möglich eingefordert, damit auch "Spezialfälle" abgedeckt sind und so ein möglichst grosse Testabdeckung gewährleistet wird.	80	
Performance	Die Performance ist trotz des neuen Designs der Lösung noch immer nicht den Erwartungen entsprechend.	Es werden POC (Proof Of Concept) in der Analyse- und Implementations-Phase erstellt, welche dies vorbeugen.	100	
Usability	Die Usability verschlechtert sich, statt besser zu werden.	Durch diverse Usability-Tests, unter anderem auch durch Feldforschungen sollte die Usability verbessert werden. Auch durch explizites Abklären mit dem Projektpartner.	100	
Prototypen	Die Prototypen gehen vollkommen in die falsche Richtung. Die Implementation der Prototypen kann nicht weiter verwendet werden.	Durch Wireframes, welche in der Analyse-Phase erstellt und mit dem Projektpartner besprochen werden, wird gewährleistet, dass die Implementation der Prototypen keine falsche Richtung einschlägt.	60	
Feldforschung (Reality-Check)	Die Feldforschung zeigt, dass die Lösung nicht den Erwartungen entspricht, da sie z.B. zu technisch ausgelegt ist, nicht domänenspezifisch, zu abstrakt, etc.	Durch User-Stories, welche vom Projektpartner aufgezeigt werden, wird dies vermieden.	60	
Daten-Analyse	Die Daten werden falsch interpretiert, bzw. verstanden. Dadurch werden sie falsch angezeigt und führen zu Verwirrung.	Die Daten werden vom Team analysiert und mit dem Projektpartner nochmals genau besprochen.	70	
Neue Erkenntnisse	Während der Implementations-Phase werden neue Erkenntnisse gewonnen, welche den aktuellen Stand der Entwicklung "über den Haufen schmeissen".		45	
Meilensteine	Meilensteine können nicht eingehalten werden, wodurch die Planung nach hinten verschoben wird und die Zeit knapp wird.	Die Anforderungen genau spezifizieren und die Arbeitsaufwände immer wieder neu schätzen, damit die Erreichung der Meilensteine gewährleistet ist. Scope im Auge behalten. Entwicklung in die Breite anstatt Tiefe.	50	

Mit der Änderung der Stossrichtung in der 9. Woche, entschieden wir uns das Risikomanagement nochmals zu überdenken und allfällige Risiken zu ergänzen. Folgend diejenigen Risiken, welche neu definiert wurden.

RISIKO	BESCHREIBUNG	VORBEUGUNG	S	E
Usability / Benutzerakzeptanz	Dieser Ansatz wird vom Benutzer nur als schwer verständlich angenommen und braucht diverse Erläuterungen, um korrekt und effizient genutzt zu werden.	Durch Abklärungen mit Stephan von der Varian und diversen Usability-Test versuchen wir dieses Risiko zu verhindern.	100	✗
Umsetzung	Bei der Umsetzung dieses Ansatzes können die "Default"-Controls von WPF nicht genutzt werden, d.h. wir müssen unsere eigene Controls bauen.	Die Controls werden evaluiert und eine Richtung eingeschlagen, welche verhindern sollte, dass dieses Risiko zum Problem wird.	50	✓
Performance	Die Datenmenge sollte kein Problem darstellen beim Laden der Daten, jedoch könnte es beim Darstellen der Daten, bzw. das Auf- und Zuklappen der Ansichten zu Performance-Probleme kommen.	Durch gezieltes "Zeichnen"/Laden der Controls wird dies vorgebeugt. Während der Entwicklung darauf achten.	90	✗
Zeit	Durch die fortgeschrittene Zeit und die Anpassung des Projektplans könnte die Zeit sehr knapp werden.	Features werden priorisiert und solche mit hoher Priorität werden zuerst implementiert, damit der Zeitplan auch eingehalten werden kann.	100	✗
Neue Erkenntnisse	Während der Implementation ergeben sich neue Erkenntnisse/Ansätze.	Neue Erkenntnisse/Ansätze werden nicht weiter verfolgt, um den Zeitplan einhalten zu können.	20	✗

Im Risiko-Matrix-Diagramm ist die Ausgangslage der Risiken ersichtlich, wie sie von uns zu Beginn eingeschätzt wurden.



Continuous Integration

Wir haben Visual Studio Online für den Continuous-Integration Prozess verwendet. Dort haben wir jeweils unseren Stand der Implementierung eingecheckt und ebenfalls automatische Builds eingerichtet. Für die Prototypen haben wir keine Unit Tests erstellt, wobei wir auch darauf verzichtet diese automatisch bauen zu lassen. Für die Implementation des RT Summary haben wir jedoch den Buildserver so eingerichtet, dass jeweils nach jedem Check-In die Solution gebuildet wird und die Unit Tests durchgeführt werden.

Analyse

Domänenanalyse

Die medizinische Domäne war für uns beide neu. Deshalb sind wir sehr dankbar für die ausführliche Einführung zu Beginn des Projektes durch unseren Projektpartner (Stephan Huber, Varian). Obwohl die Einführung sehr gut war, mussten wir uns nochmals ein Bild verschaffen, wie die einzelnen Begriffe/Schritte zusammenhängen.

Unsere Applikation ist ein Teil von einem gesamten Informationssystem, wobei es grundsätzlich um die Strahlentherapie geht. Deshalb holen wir bei der Beschreibung ein bisschen aus, wie wir dies auch bei der Arbeit gemacht haben, damit auch der Zusammenhang klarer wird.

Strahlentherapie (Radiotherapy)

“ Strahlentherapie (Radiotherapy) ist das medizinische Fachgebiet, das sich mit der medizinischen Anwendung von ionisierender Strahlung auf den Menschen [...] beschäftigt, um Krankheiten zu heilen oder deren Fortschreiten zu verzögern. [...] [WSB] ”

Onkologie (Oncology)

“ Als Onkologie [...] bezeichnet man die Wissenschaft, die sich mit Krebs befasst. Im engeren Sinne ist die Onkologie der Zweig der inneren Medizin, der sich der Prävention, Diagnostik, konservativen Therapie, [...] widmet. [WONK] ”

Clinical Workflow

Der Clinical Workflow zeigt auf welche Schritte ein Patient durchläuft, wenn er in einer onkologischen Behandlung ist.



CLINICAL WORKFLOW		INSTANCE
Register	Patient Registrierung - Personenangaben erfassen Diagnose/Befund erfassen	Administration
CT Computed Tomography	Integration in DICOM Worklist	RT Stuff
Plan	Planung der Bestrahlung	Medizinischer Physiker
Prepare	Vorbereitung der Bestrahlung	RT Stuff
Approve	Bestätigung der Vorbereitung durch den medizinischen Physiker	Medizinischer Physiker
Schedule	Behandlungen planen	RT Stuff
Treat	Definierte Behandlung durchführen	RT Stuff
Review	Begutachtung und Besprechung der Behandlung	Doctor

Das RT Summary befindet sich in den Teilschritten Approve, Treat und Review. Wie in der Tabelle ersichtlich, gibt es grundsätzlich drei verschiedene Instanzen, welche mit dem RT Summary arbeiten. Diese Instanzen sind alle an unterschiedlichen Ansichten/Daten interessiert. Wodurch die Möglichkeit bestehen würde, jeweils verschiedene Ansichten, bzw. Einstiegspunkte zu definieren. Dies ist nicht Teil unserer Aufgabe, aber als weiteres Feature denkbar.

Terminologie

Die Terminologie hält sich in Grenzen, jedoch gibt es essentielle Begriffe, welche für das weitere Verständnis sehr wichtig sind. Diese haben wir in der folgenden Tabelle kurz erläutert.

BEGRIFF	BESCHREIBUNG
Course / Kurs	Der Kurs beinhaltet alle Bestrahlungsereignisse, welche für das therapeutische Ziel relevant sind. Ebenfalls werden die Pläne in einem Kurs gruppiert. Diese Pläne können sequentiell oder auch parallel durchgeführt werden um das therapeutische Ziel auch effizient zu erreichen.
Plan	Ein Plan enthält die zu behandelnden Feld. Wobei ein Plan in verschiedene Fractions fraktioniert wird. Im Normalfall dauert die Behandlung eines Planes (oder auch mehrere parallel) 30 Tage. Dies jedoch mit Pausen und Regenerationszeiten. Dadurch ergibt sich, dass ein Plan normalerweise über zwei Monate durchgeführt wird.
Treatment Session / Behandlung	Eine Behandlung gruppiert alle Fractions, welche an einem Tag behandelt werden. Diese sind auch Plan übergreifend. Eine Behandlung zeigt effektiv alle Felder, welche an einem ausgewählten Tag behandelt werden, an.
Fraction	Die Fraction enthält nur Informationen über einen Plan und dessen Feldern, welche an einem Tag behandelt werden. Somit hat ein Plan im Normalfall 30 Fractions.
Field / Feld	Ein Feld beinhaltet alle notwendigen Parameter und die Dosis, welche für die Bestrahlung benötigt werden. Diese Parameter können während der Behandlung auch angepasst werden, um auf Ausnahmen/Änderungen zu reagieren.
Dose / Dosis	Die Dosis wird grundsätzlich über einen ganzen Plan, bzw. Kurs definiert und danach auf die einzelnen Felder oder auch Fractions aufgeteilt.

Anforderungsanalyse

Zu Beginn der Bachelorarbeit haben wir unter anderem eine vereinfachte Datenschnittstelle zum aktuellen Stand der Daten erhalten. Obwohl die Datenschnittstelle vereinfacht war, mussten wir Zeit investieren diese genauer zu analysieren, um uns einen Überblick zu verschaffen (Siehe Kapitel Datenmodell). Des Weiteren wurden zu Beginn ebenfalls einige Anforderungen definiert.

Anforderungen

PERFORMANCE

Initial Load	Large Patient < 5 second Medium Patient < 3 second Small Patient < 1 second
User Interaction	Scrolling, selection, etc. -> Immediate

Der Scope der Anwendung wurde ebenfalls zu Beginn besprochen. Der Scope definierte die Anforderungen an die neu zu erstellende Applikation.

SCOPE

Course Timeline	Zeigt die Kurse eines Patienten in einer Zeitachse an.
Involved Plans	Die Pläne eines Kurses werden angezeigt.
Past Treatments (Partial Treatments)	Wie aus der Domänenanalyse ersichtlich ist, beinhaltet ein Plan mehrere Behandlungen. Diese müssen angezeigt werden. Eine Behandlung kann partial sein, wenn die Behandlung aus diversen Gründen nicht vollumfänglich durchgeführt wurde. Diese müssen speziell gekennzeichnet werden.
Future Treatments	Da Behandlungen im Voraus geplant werden können, müssen diese ebenfalls speziell gekennzeichnet werden.
Field Parameters	Die Anzeige der relevanten Parameter soll konzipiert und gemockt (vorgetäuscht) werden, da diese Daten in den Testdaten nicht vorhanden sind.
Imaging activities	Während einer Behandlung können diverse Bildaufnahmen gemacht werden. Diese werden danach durch den Arzt approved (genehmigt). Die genehmigten Images sollen im RT Summary dargestellt werden. Der Genehmigungsprozess ist nicht Teil unserer Aufgabe.
Exceptions / Overrides	Während des Treatments können auch Ausnahmen/Änderungen vorkommen. Diese müssen ebenfalls klar ersichtlich dargestellt werden.

Feldforschung

Uns war es wichtig, dass wir mit unserem Stand der Analyse auf die Vorstellungen und Wünsche des Projektpartners eingehen können. Des Weiteren wollten wir auch eine optimale Lösung entwickeln, welche den Standards des Projektpartners entspricht. Hierfür hatten wir die Möglichkeit zwei Usability Workshops durchzuführen.

1. Usability Lab

Wir konnten beim ersten Usability Workshop (Woche 6) von Fachleuten profitieren. Von Seitens des Projektpartners konnten wir ein Interview mit einem Usability Experten (Beat Markwalder, Varian Medical Systems Imaging Laboratory Baden) und einem Dosimetrist (in der Rolle als Arzt, Marco Lessard, Varian Medical Systems Canada) führen. Bei diesem Usability Workshop wollten wir diverse Abklärungen mit den entsprechenden Personen treffen. Primär war uns wichtig herauszufinden, ob wir uns mit den bis dahin erstellten Prototypen in die richtige Richtung bewegten oder ob diverse Ansätze geändert werden müssen.

Unsere Traktanden für diesen Workshop waren folgende:

- Prototypen vorstellen und Feedback einholen (Usability-Experte, Dosimetrist, sowie auch von Stephan Huber)
- Bestmögliche Usability für das RT Summary
- Diverse Usability-Prinzipien besprechen
- Weiteres Vorgehen besprechen

Nach einer kurzen gemeinsamen Analyse des bestehenden RT Summarys präsentierten wir unsere drei Prototypen: Fahrplan, Horizontal-Timeline, Horizontal-Summary-Timeline (siehe Lösungsvarianten). Diese kamen bei allen anwesenden Workshop Teilnehmern sehr gut an. Nach langen und konstruktiven Diskussionen entschieden wir gemeinsam den Prototyp „Fahrplan“ weiter zu verfolgen.

Auswertung Workshop mit Usability-Experte

Das Wichtigste was wir aus dem Workshop mit dem Usability-Experten mitgenommen haben, war die Art der Zeit-Anzeige, welche in Ansicht des Usability-Experten horizontal verlaufen sollte. Des Weiteren wurde beschlossen, dass ein horizontales Scrolling nicht intuitiv ist und deshalb darauf geachtet werden soll, dass wenn Scrolling nötig ist, dieses vertikal verläuft.

Auswertung Workshop mit Dosimetrist/Arzt

Nebst der Präsentation unserer Prototypen haben wir beim Workshop mit dem Dosimetristen ein Card-Sorting eingesetzt um herauszufinden, welche Angaben für einen Arzt wichtig sind. Das Ergebnis ist folgend zu sehen, wobei die Priorität von links nach rechts verläuft, d.h. höchste Priorität hat die Angabe des Courses, danach die Dosis, etc.

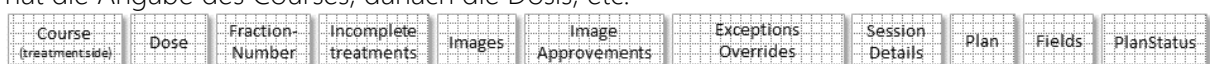


Abbildung 1: Card-Sorting Auswertung

2. Usability Lab

Beim zweiten Usability Workshop (Woche 13) waren unser Betreuer, Stephan Huber und der Usability Experte (Beat Markwalder) anwesend. Damit die Integration ins bestehende Informationssystem auch reibungslos verläuft, wollten wir frühzeitig einen weiteren Usability Workshop durchführen. Dieser hatte unter anderem zum Ziel, dass unser Control auch dem Style-Standard der Varian Anwendungen entspricht.

Die Traktanden dieses Workshops haben wir auf folgende eingeschränkt:

- Aktueller Stand der Arbeit
- Demo des aktuellen Entwicklungsstandes
- Weiteres Vorgehen in Bezug auf Styling und Integration

Der aktuelle Stand kam bei allen anwesenden Parteien gut an und wurde mehrmals gelobt. Nichts desto trotz wurde eine konstruktive Diskussion geführt. Viele Features, welche wir geplant haben, wurden durch den Projektpartner bestätigt, wobei auch neue Features gewünscht worden sind. Folgend eine Auflistung der gewünschten Features, bzw. Features, welche während diesem Usability Lab erarbeitet wurden:

- Wenn ein Plan zugeklappt ist -> Batches für Ausnahmen/Bildaufnahmen anzeigen
- Tick bei approved Bildaufnahmen
- Wochentag anzeigen z.B. **FR**, 16.05.2014
- Bildaufnahmen in Session-Details anzeigen
- Icons verwenden zusätzlich zur Farbcodierung
- Konzept für Anzeige, dass ein Plan nicht ganz offen ist
- Zeitachse anpassen (Woche alternierend einfärben, Monate anzeigen)
- ScheduledPlans ohne Datum in der Zeitachse anzeigen
- Minimum-Size für Plan (wenn zugeklappt/offen)
- Bildaufnahmen-Anzeige und Approvment auf der rechten Seite
- Datum und Startzeit bei Session-Details anzeigen
- Session-Details farblich markieren bei Selektion
- Bildaufnahmen/Ausnahmen Zusammenfassung (nur Icons und Anzahl)
- Session-Details ausblenden, wenn Pläne zugeklappt
- Ausnahmen pro Tag anzeigen und Border von Bildaufnahmen und Ausnahmen nicht schwarz
- FuturePlans Farbe anpassen gleiche Farbe wie „normale“ Pläne
- Icons für Dauer und relative Zeit bei Session-Details (Stoppuhr und Uhr)
- Übersetzung -> BAML und Resourcefile für ViewModel

Die Umsetzung dieser Features, bzw. Anpassungen werden im Kapitel Resultat weiter beschrieben.

Lösungsvarianten

Wir hatten uns bereits nach der Einführung in die Domain Gedanken über die Darstellungsmöglichkeiten der Daten gemacht.

Nach der Verfeinerung der Daten-Ausgangslage, welche wir vom Industriepartner erhalten haben, ging das Brainstorming weiter. Wir gingen mit verschiedenen Ansätzen an die Ideensammlung für das neue Usability-Konzept. Einerseits wollten wir unabhängig voneinander Ideen suchen, um dann diese gemeinsam zu verfeinern, andererseits hatten wir auch Ansätze von A-Z gemeinsam besprochen, wobei wir alle Ansätze kritisch hinterfragten.

Unsere Lösungsvarianten beinhalten einige Ansätze, welche teilweise ähnlich, teilweise aber auch in Bezug auf die Darstellungen der Daten sehr unterschiedlich sind. Wir erstellten zu allen Ansätzen Wireframes mit dem Tool „Balsamiq Mockups“, im PowerPoint oder auch als C# Prototyp. Somit konnten wir auch einfacher mit dem Industriepartner und uns unserem Betreuer die Stossrichtung bestimmen, da wir eine grosse Anzahl an Varianten auferlegt haben.

Schlussendlich sind wir auf elf Varianten gekommen, welche vielfach bereits relativ detailliert durchdacht sind, jedoch nur zwei davon für die Wahl der Umsetzung in die engere Wahl gekommen sind. In Absprache mit allen Beteiligten wurde schliesslich eine der Varianten zur Implementation und Detailausarbeitung gewählt.

Hier ein Überblick über alle Varianten, welche auch im Verlauf dieses Kapitels im Detail aufgeführt sind.

VARIANTE	BESCHREIBUNG
Outlook	Wie der Name schon verrät, handelt es sich um eine Darstellung wie aus dem Outlook-Kalender bekannt.
Timeline-Stack	Dieser Ansatz baut die Ansicht vertikal auf, wobei immer tiefer ins Detail gezoomt wird.
Vertical-Timeline	Die Zeit wird hier vertikal dargestellt und bietet die Möglichkeit in einzelne Tage/Monate hinein zu zoomen für mehr Details.
Metro	Dies sollte dem Style einer Microsoft Metro App entsprechen und auch deren Navigation brauchen um zu den Details zu gelangen.
Fahrplan	Hier werden die Behandlungen mit Start/Ziel versehen und die Behandlung verläuft einem Fahrplan ähnlich.
Horizontal-Timeline	In diesem Ansatz wird die Zeitachse, so wie auch die Daten horizontal dargestellt.
Horizontal-Summary-Timeline	Ähnlich wie beim vorhergehenden werden hier jedoch die Daten zusammengefasst und dargestellt.
Tree	Die Daten werden hierarchisch als Baumstruktur dargestellt, so dass der Benutzer immer tiefer in die Hierarchie navigieren kann.
2D	Der Ansatz zeigt in 2-Dimensionen alle nötigen Informationen an und gruppiert diese nach Monat.
2D++	Dies ist die Erweiterung von 2D, denn sie gruppiert die Daten nach Plänen und nicht nach Monat.
Simplified Vertical Timeline	Der Ansatz verbindet einige Aspekte aus den vorhergehenden Ansätzen und zeigt die Daten nach Plänen gruppiert in einer vertikalen Zeitachse an.

Outlook

Die Grundidee hinter dem Outlook-Ansatz ist, die Behandlungen in einer für Laien sehr verständlichen Form anzuzeigen. Dazu haben wir uns an der Terminübersicht aus dem Outlook orientiert. Bei einem Klick auf eine Behandlung werden die Details (welche Felder, welche Bildaufnahmen) in einem Panel unterhalb angezeigt. Dieser Ansatz wurde bereits zu Beginn wegen der geringen Platzausnutzung verworfen.

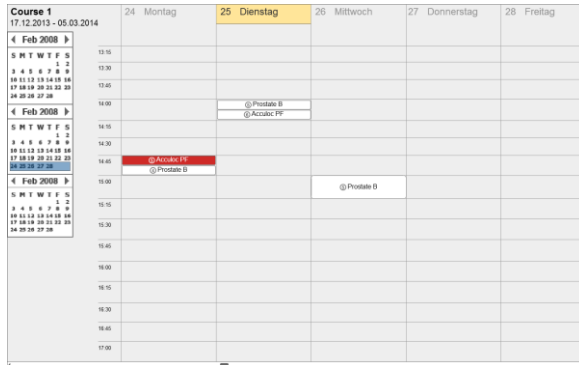


Abbildung 2: Plan-/Behandlungsübersicht

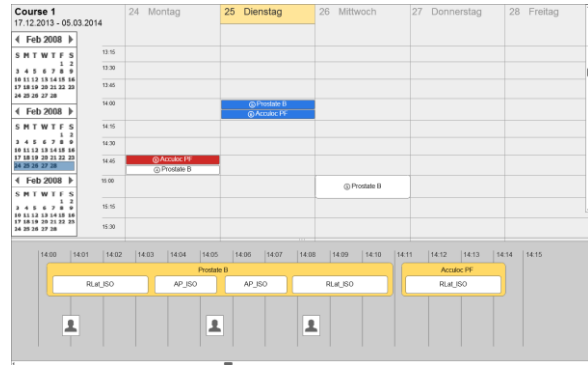


Abbildung 3: Behandlungsinformationen

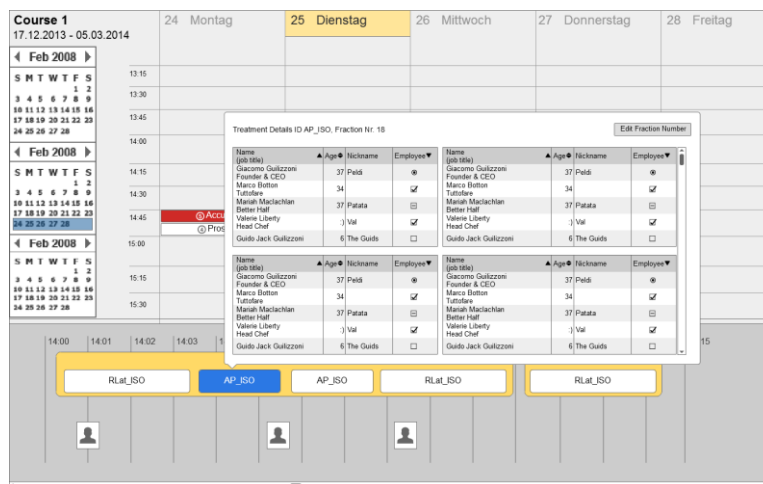


Abbildung 4: Feld-Details

POSITIV (NÜTZLICH)

Stärken

- + Bekannt durch Kalender/Outlook
- + Aufgeräumte Oberfläche

Chancen

- + Gutes Lazy-Loading möglich
- + Microsoft Guidelines benutzbar

NEGATIV (SCHÄDLICH)

Schwächen

- Sehr viel Leerraum bei wenig Behandlungen
- Überblick schwer zu verschaffen, da man zu sehr im Detail ist
- Häufiger Wechsel zwischen Tagen nötig

Bedrohungen (Risiken)

- Keine Übersicht über Alles
- Kalender-Metapher könnte fehlinterpretiert werden

Timeline-Stack

Die Idee hinter dem Timeline-Stack-Ansatz ist, die ganzen Informationen auf verschiedene Zeitachsen mit verschiedenen Granularitäten aufzuteilen. So werden in der ersten Zeitachse die Kurse, danach die Pläne usw. bis zu den Details angezeigt. Auf jeder Ebene kann eine Auswahl getroffen werden, die dann die Zeitachse auf dem nächsten Level entsprechend aktualisiert.

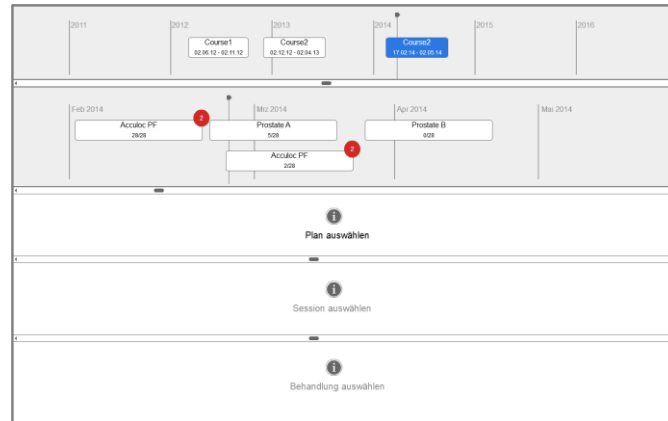


Abbildung 5: Auswahl eines Kurses -> Anzeige der Pläne

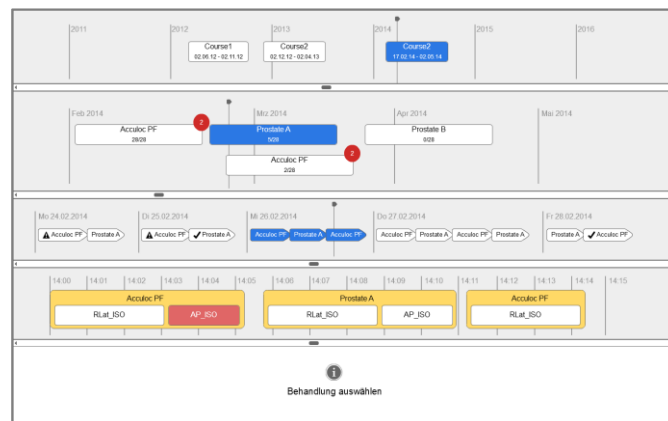


Abbildung 6: Zoom-In für mehr Informationen

POSITIV (NÜTZLICH)

Stärken

- + Nicht zu viele Infos aufs Mal, Schritt-für-Schritt Aufbau (Kurs->Plan->Tag->Behandlung->Details)
- + Konsistent (jeder Teil-Schritt ist auch eine Zeitachse)
- + Ausnahmen sind auf High-Level Ebene ersichtlich
- + Zeitlicher Ablauf gut ersichtlich

Chancen

- + Gute Wiederverwendbarkeit von GUI-Komponenten (Timeline-Control)
- + Teilschritte müssen mit weniger Daten umgehen/anzeigen, da bereits durch übergeordnete Auswahl gefiltert

NEGATIV (SCHÄDLICH)

Schwächen

- Viele Schritte nötig um Details einzusehen
- Fixe Struktur (Kurs->Plan->Tag-> Behandlung-> Details)
- Bei vielen Daten in der Tagesansicht -> Viel Scrolling nötig

Bedrohungen (Risiken)

- Wartezeit bei jedem Teilschritt möglich
- Benutzer weiss nicht, wo er sich aktuell befindet

Vertical-Timeline

Beim Vertical-Timeline-Ansatz haben wir uns die Idee einer um 90 Grad gedrehten Zeitachse angeschaut. Das Kurs Start- und Enddatum ist jederzeit sichtbar. Dazwischen wird entlang der Zeitachse gescrollt. Die Pläne sind jeweils rechts von den Tagen, die Ausnahmen links davon angeordnet. Bei der Auswahl eines Tages werden die Felder etc. direkt inline angezeigt.

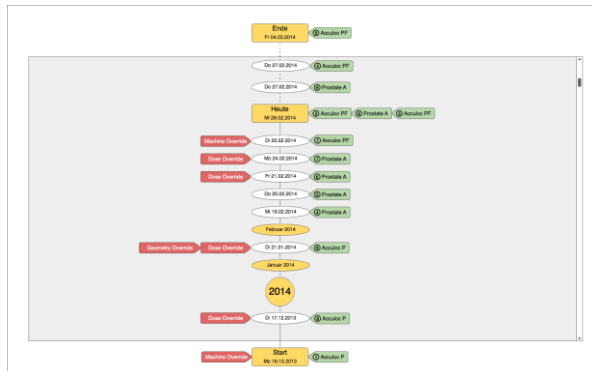


Abbildung 7: Vertikale Zeitachse mit Detailinformationen

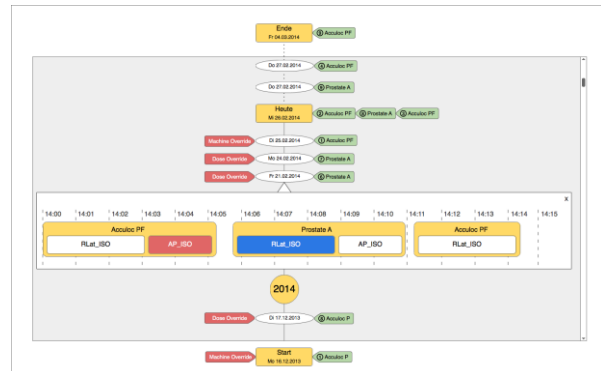


Abbildung 8: Tagesauswahl mit mehr Informationen

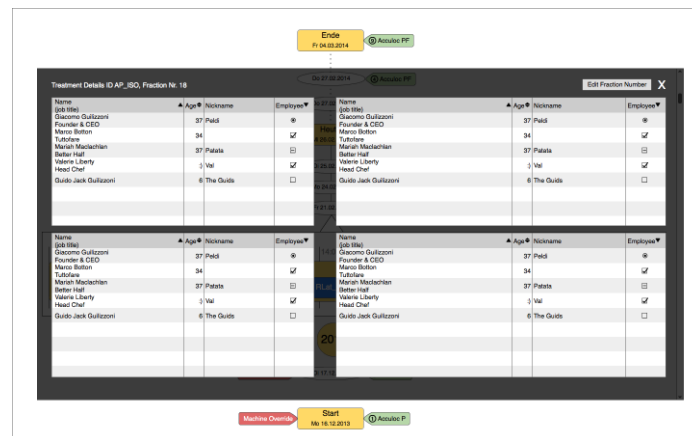


Abbildung 9: Detail-Informationen zur Behandlung

POSITIV (NÜTZLICH)

Stärken

- + Schnelle Übersicht über die nächsten/letzten Behandlungen
- + Ausnahmen schnell ersichtlich (immer Links)
- + Behandlungen direkt an der Achse angeschrieben.
- + Zeitpunkt des Kurses (Start- und Enddatum) jederzeit sichtbar, fixiert beim Scrolling
- + Anpassbarkeit (Bereiche Ein/Ausklappen)

Chancen

- + Relativ simpler Aufbau der View (Flache Liste anstatt Grid)

NEGATIV (SCHÄDLICH)

Schwächen

- Platzverschwendung in der Breite (evtl. Möglichkeit für Side-by-Side Vergleich)
- Keine Zusammenfassung von alten Daten

Bedrohungen (Risiken)

- Alle Daten müssen geladen sein, nachladen beim Scrolling unschön
- Übersicht der Pläne schlecht sichtbar

Metro

Der Metro-Ansatz entspricht vom Design her den aktuellen Microsoft-Store Apps. Ein Windows 8 Benutzer könnte sich dadurch relativ schnell einarbeiten. Durch die bekannten Features und die Erfahrung mit der Navigation durch die Microsoft-Store Apps kann der Benutzer zielorientiert und schnell seine Daten anzeigen lassen.

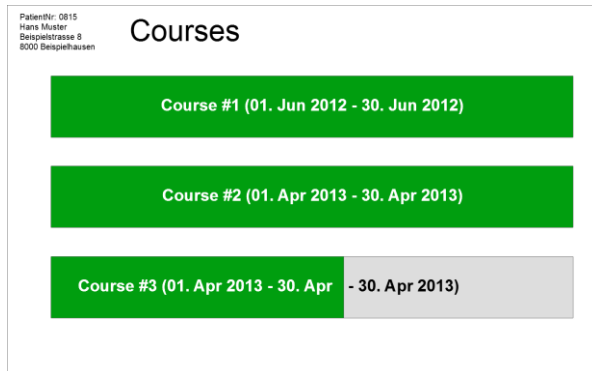


Abbildung 10: Übersicht über Kurse eines Patienten

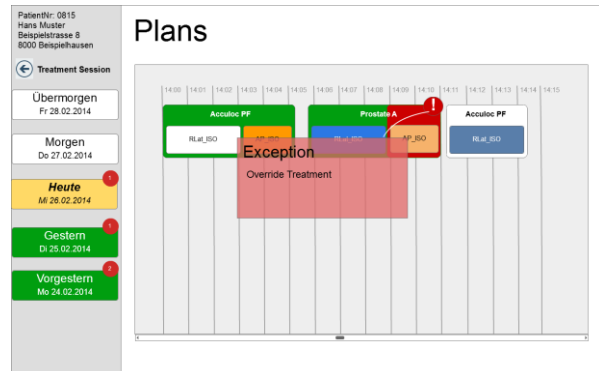


Abbildung 11: Tages-Ansicht der Behandlungen



Abbildung 12: Detail-Informationen einer Behandlung

POSITIV (NÜTZLICH)

Stärken

- + Die einzelnen Views zeigen nicht zu viele Daten aufs Mal an.
- + Gute Benutzerführung durch Schrittweise Navigation

Chancen

- + Guidelines für Microsoft-Store Apps können berücksichtigt werden
- + Erfahrene Benutzer finden sich schnell zu recht

NEGATIV (SCHÄDLICH)

Schwächen

- Hin und her springen zwischen Views nötig.
- Fixer Ablauf der Views

Bedrohungen (Risiken)

- Benutzer verirrt sich durch das häufige Klicken
- Keine Gesamtübersicht in den einzelnen Ansichten
- „Metro“-Style für Benutzer unübersichtlich

Fahrplan

Beim Fahrplan-Ansatz wollten wir, wie der Name schon aussagt, dass die Behandlungen ähnlich einem Fahrplan dargestellt werden. Dies bedeutet, dass die Behandlungen zeitlich vertikal angezeigt werden und jeweils gekennzeichnet sind, welche in der Vergangenheit und welche in der Zukunft liegen. In der Horizontalen wollen wir die Pläne anzeigen, wobei diese jeweils nur dann ersichtlich sind, wenn auch Behandlungen im selektierten Zeitfenster vorhanden sind.

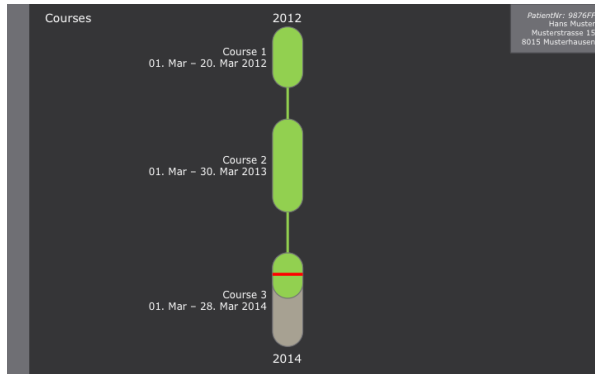


Abbildung 13: Kurs-Übersicht des Patienten

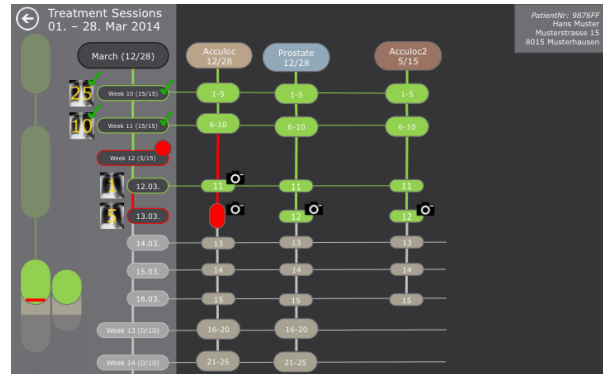


Abbildung 14: Zoom-In für mehr Informationen einer Behandlung

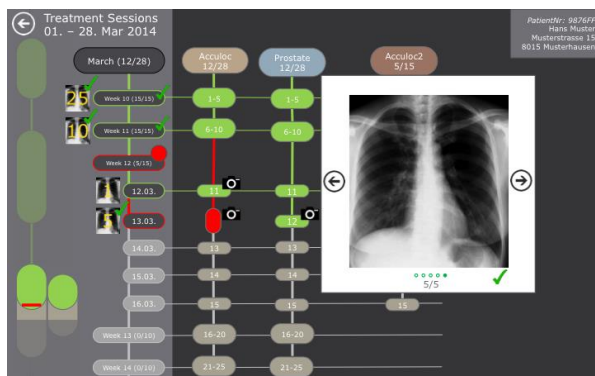


Abbildung 15: Bildaufnahmen-Genehmigung

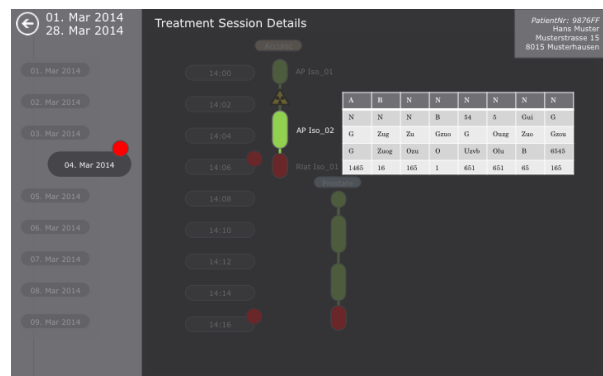


Abbildung 16: Detail-Informationen einer Behandlung

POSITIV (NÜTZLICH)

Stärken

- + Kurse übersichtlich auf der Startseite
- + Ausnahmen in einem Kurse sind auf der ersten Seite ersichtlich
- + Zeitlicher Ablauf der Kurse ersichtlich
- + Ausnahmen sind klar ersichtlich (z.B. Behandlung)

Chancen

- + Zeitraster für eine klarere Übersicht über Behandlungen auf der Startseite
- + Weniger ist manchmal mehr, die Views sind nicht überladen
- + GUI ist modern und weist bekannte Elemente (z.B. „Batches“ von Apple)

NEGATIV (SCHÄDLICH)

Schwächen

- Anzahl Behandlungen nicht so klar ersichtlich
- Platzauslastung gering (Startseite)
- Orientierung, welche Behandlung zu welchem Datum gehört nicht ganz klar (horizontale „Augenwanderung“)

Bedrohungen (Risiken)

- User Szenario (View-Abfolge) fix definiert, lässt wenig Spielraum
- Overlay kann für Verwirrung sorgen
- Entwicklung: sehr modernes GUI, jedoch auch sehr aufwändig

Horizontal-Timeline

Dieser Prototyp ist während der Prototypen-Implementierung entstanden. Eigentlich ist es eine Abwandlung des Fahrplan Ansatzes, einfach eine horizontale Anordnung. Die Pläne sind in der Mitte angeordnet. Mit diesem Prototyp wurde auch untersucht wie intuitiv sich das horizontale scrollen anfühlt. Das horizontale Scrollen hat sich beim Test mit diesem Prototyp nicht bewährt. Da das Mausrad nicht für horizontales Scrollen konzipiert wurde, führt dies eher zu Verwirrung. Dieser Ansatz wurde daher nicht weiter verfolgt.

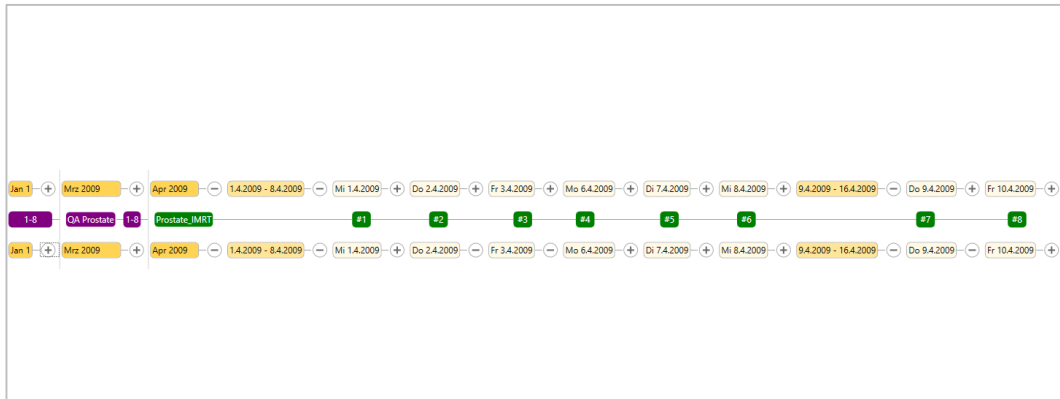


Abbildung 17: Monat/Woche aufgeklappt

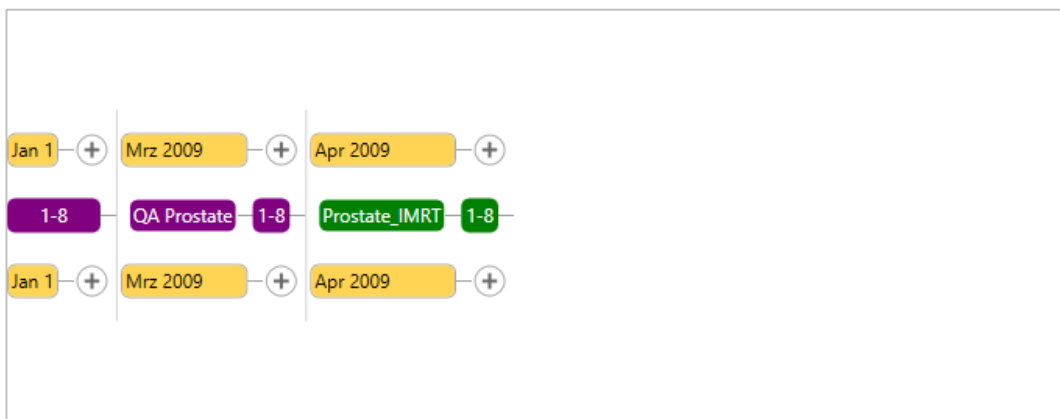


Abbildung 18: Zugeklappte Ansicht

POSITIV (NÜTZLICH)

Stärken

- + Kompakte Darstellung
- + Horizontaler Planverlauf gut sichtbar

Chancen

- + Ansicht nicht überladen
- + Gruppierung von Fractions
- + Collapse von alten Daten

NEGATIV (SCHÄDLICH)

Schwächen

- Scrolling horizontal mit Mausrad nicht intuitiv
- Platzverschwendung bei wenig parallelen Plänen

Bedrohungen (Risiken)

- Übersicht der verschiedenen Pläne schlecht sichtbar
- Zu viele Zeit-Ebenen (Monat/Woche/Tag), welche einzeln aufgeklappt werden müssen

Horizontal-Summary-Timeline

Dieser Prototyp ist eine Weiterentwicklung des Timeline-Stack-Ansatzes. Die Pläne werden jeweils ober- bzw. unterhalb (bei parallelen Plänen) des Datums angezeigt. Das Datum ist immer zentriert in der Mitte. Wurde als Prototyp für das Usability Lab 1 implementiert.



Abbildung 19: Monat/Woche aufgeklappt

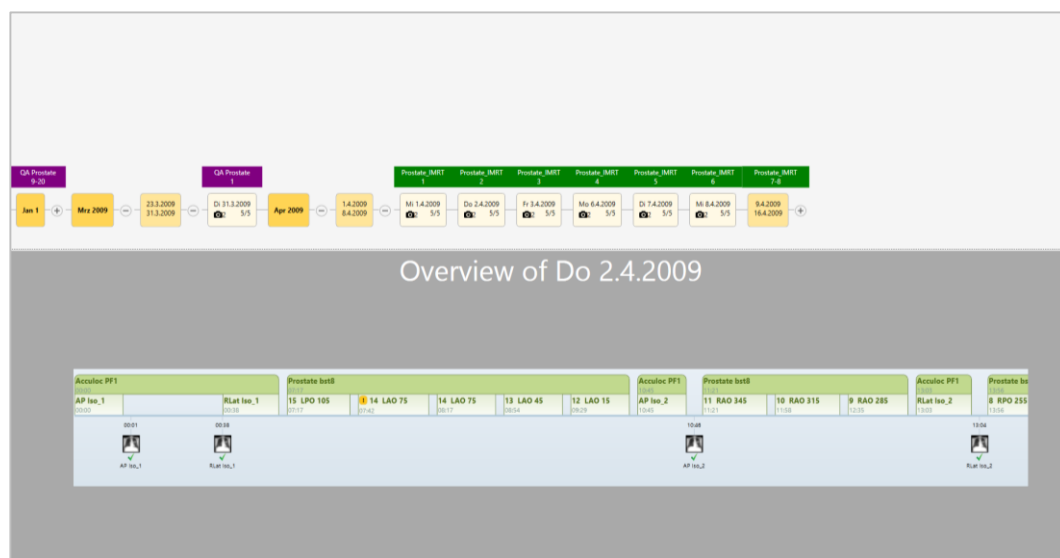


Abbildung 20: Tag ausgewählt

POSITIV (NÜTZLICH)

Stärken

- + Kompakte Darstellung
- + Datum jederzeit im Blick

Chancen

- + Gruppierung von Fractions
- + Collapse von alten Daten

NEGATIV (SCHÄDLICH)

Schwächen

- Scrolling horizontal mit Mausrad nicht intuitiv (Mausrad ist für vertikales Scrolling ungeeignet)
- Redundante Informationen (Plan-Name)

Bedrohungen (Risiken)

- Gestackte Pläne sind unübersichtlich
- Farbe pro Plan gefährlich, bei vielen Plänen (Farbwiederholung etc.)

Tree

Beim Tree-Ansatz wurden wir von einer hierarchischen Struktur wie z.B. beim Windows-Explorer inspiriert. Die Idee war hier, dass wir die Daten jeweils auf Kurs-Ebene gruppieren und dadurch auch Informationen, welche nicht relevant sind ausblenden. Durch Auf- und Zuklappen der Informationen gelangt der Benutzer zu den gewünschten Informationen. Auf jeder Ebene würden wir die essentiellen Daten jeweils zusammenfassen, wodurch der Benutzer auch bereits auf einer hohen Ebene ein Bild davon machen kann, was in dieser Zeitspanne geschehen ist.

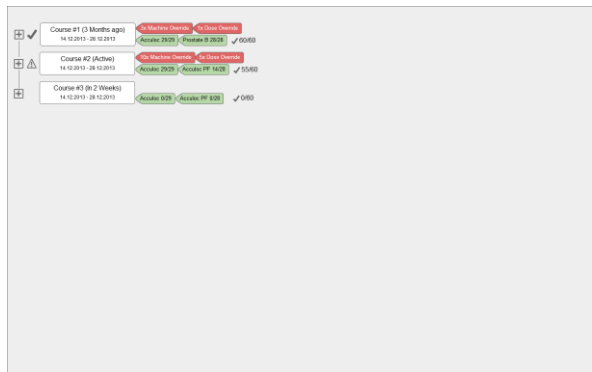


Abbildung 21: Kurs-Übersicht eines Patienten

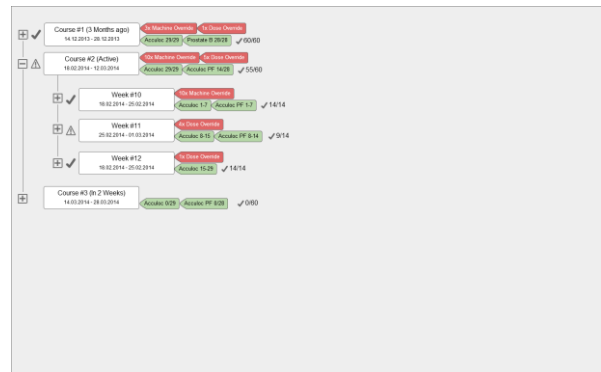


Abbildung 22: Hierarchie der Daten als Baum dargestellt

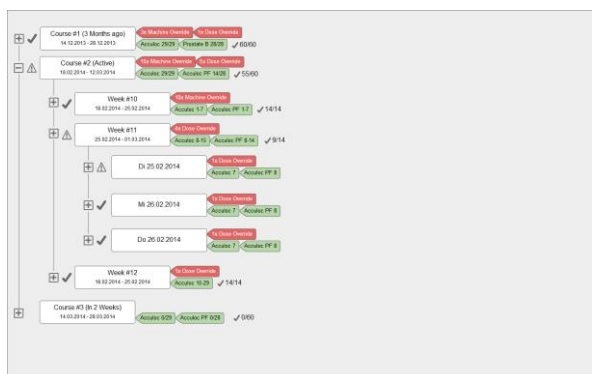


Abbildung 23: Tiefer in die Hierarchie für mehr Informationen

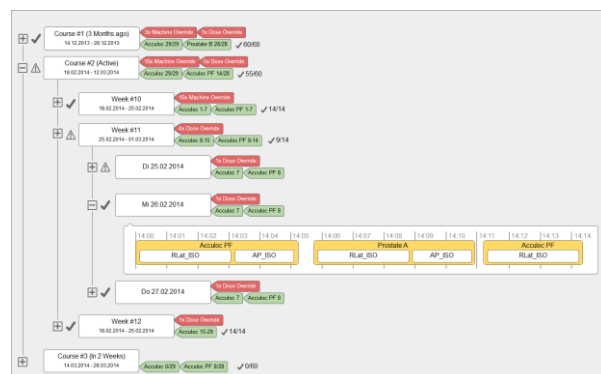


Abbildung 24: Behandlungs-Details

POSITIV (NÜTZLICH)

Stärken

- + Nicht relevante Daten können ausgeblendet werden, übrig bleibt nur die Zusammenfassung.
- + Viel Platz für die Anzeige von Detailinformationen

NEGATIV (SCHÄDLICH)

Schwächen

- Das hierarchische Denken kommt bei Laien nicht gut an
- Um auf die Tagesebene zu kommen, benötigt ein Benutzer zu viele Klicks, was sehr verwirrend sein kann
- Ein Tagesvergleich ist nur schwer möglich und nicht sehr einfach anzuzeigen, da alle benötigten Tage aufgeklappt werden müssen

Chancen

- + Gruppierung von Fractions
- + Einheitliche hierarchische Darstellung

Bedrohungen (Risiken)

- Der Ansatz wird von den Benutzern nicht verstanden
- Zu viele Wiederholungen durch hierarchische Darstellung

2D++

Dem Namen kann entnommen werden, dass dies eine Erweiterung des oben beschriebenen Ansatzes 2D ist. Nach Rücksprache mit dem Industriepartner (Stephan) haben wir den 2D-Ansatz dahingehend angepasst, dass statt einer zeitlichen Gruppierung eine Top-Level Gruppierung nach Plänen angezeigt wird. Diese enthält Informationen über Pläne, welche parallel stattfinden. Ebenfalls wurde hier die Aufteilung auf die Wochen weggelassen, womit es nur 2 Ebenen gibt, zwar die Planübersicht und danach direkt die Tage der Pläne. Die Anzeige der Detailinformationen eines Tages bleiben jedoch gleich als eigene Ansicht auf der rechten Seite.

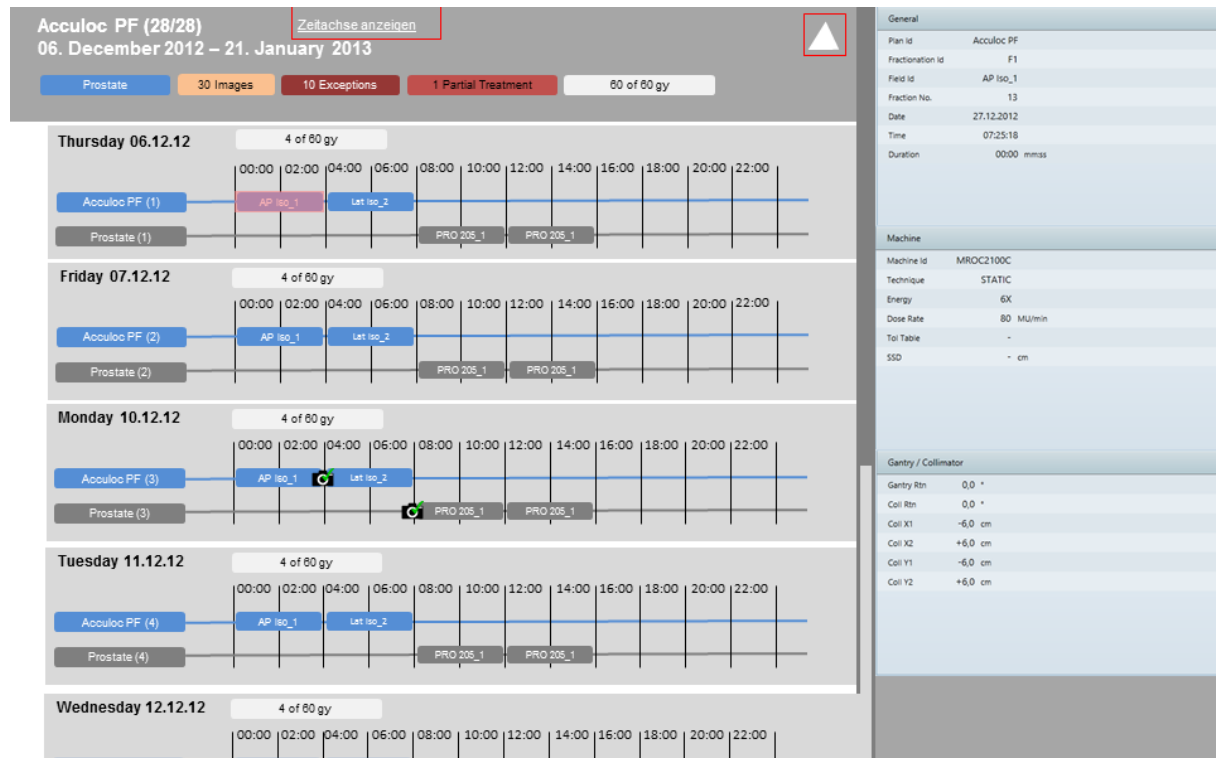


Abbildung 27: Aufgeklappte Ansicht eines Plans mit der Tagesansicht

POSITIV (NÜTZLICH)

Stärken

- + Schnelle Übersicht über die Pläne
- + Gewünschte Detailinformationen stehen rasch zur Verfügung
- + Durch die Möglichkeit auf einen Tag zu springen, ist sehr schnelles navigieren möglich

Chancen

- + Viel Platz vorhanden für allfällige Änderungen in der Informationsübersicht
- + Beliebige Erweiterbarkeit der Labels z.B. Gewicht, Dose etc.

NEGATIV (SCHÄDLICH)

Schwächen

- Benutzer verirrt sich, da viele Daten auf einmal aufgeklappt werden
- „Boxen“-Ansicht könnte bei den Benutzern zu Verwirrung führen

Bedrohungen (Risiken)

- Werden Pläne in Zukunft viel länger als 30 Tage, dann müsste in der Horizontalen gescrolled werden
- Plan wird mehrmals angezeigt

Simplified-Vertical-Timeline

Dieser Ansatz bezieht sich auf den oben genannten Fahrplan-Ansatz. Er ist jedoch vereinfachter dargestellt und zeigt die Daten auf Top-Level gruppiert nach Plänen an. Hier verläuft die Zeit vertikal, wobei die Pläne je nach freiem Platz angeordnet werden. Dies bedeutet, wenn Pläne parallel verlaufen, dann werden diese horizontal nebeneinander angezeigt. Somit baut sich die Anzeige von links nach rechts auf in der Reihenfolge der anzuzeigenden Pläne. Um an die Detailinformationen, wie z.B. Fractions oder Tagesansichten zu gelangen, muss der Benutzer den Plan aufklappen. Hierbei werden alle Fractions angezeigt, welche parallel zu dem ausgewählten Plan verlaufen. (siehe Bild) Beim Auf- bzw. Zuklappen wird ebenfalls die Granularität der Zeitanzeige auf der linken Seite verändert.

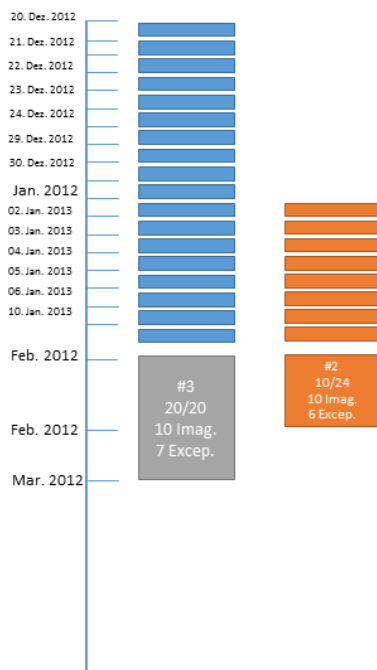


Abbildung 28: Aufgeklappter Plan mit aufgeklappten Fractions des parallelen Plans

POSITIV (NÜTZLICH)

Stärken

- + Gesamtübersicht klar
- + Zeitliche Abfolge der Pläne/Kurse gut ersichtlich.
- + Bei Standard-Plänen gibt es genügend Platz um Zusammenfassungen anzuzeigen.
- + In der Detailansicht lassen sich Dosen, Ausnahmen, Bildaufnahmen, etc. gut anzeigen.

Chancen

- + Sehr hohe Benutzerakzeptanz
- + Scrolling intuitiv

NEGATIV (SCHÄDLICH)

Schwächen

- Bei wenigen Plänen, sehr viel Leerraum, welcher nicht genutzt wird.
- Verwirrend, dass beim Auf-/Zuklappen nur gewisse Teile von parallelen Plänen geöffnet oder geschlossen werden.
- Bei zu vielen Plänen ist ein scrollen horizontal und vertikal möglich.

Bedrohungen (Risiken)

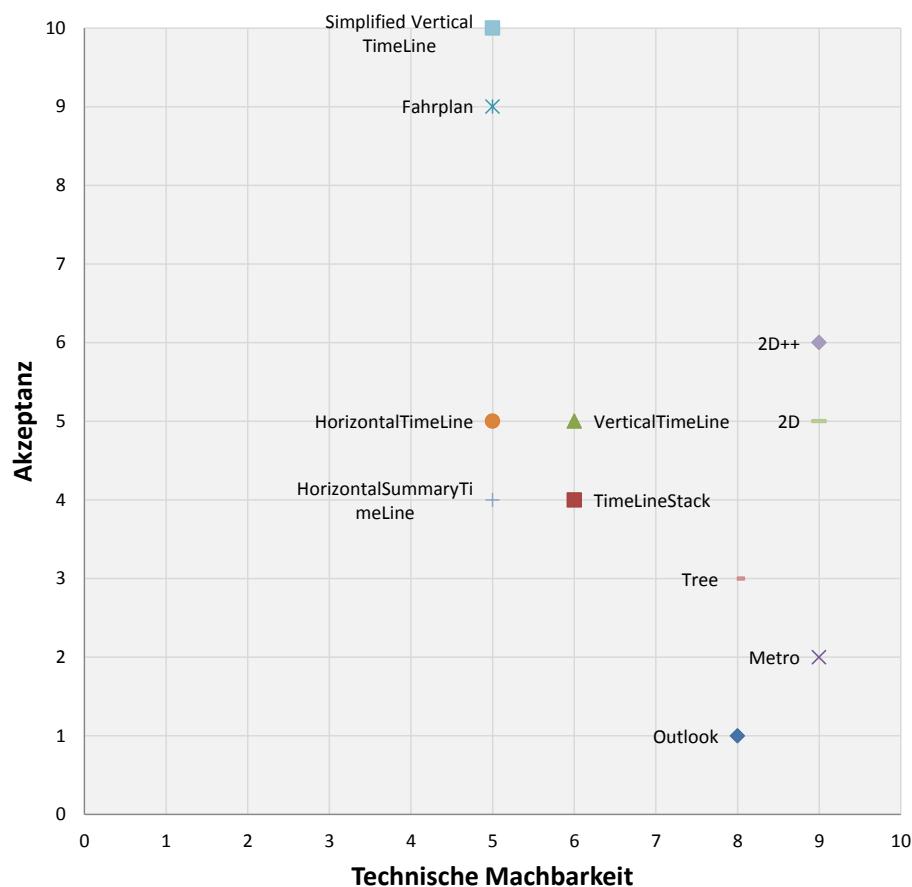
- Sehr aufwendige Implementierung. Standard-Controls vom .NET Framework ungeeignet.

Gegenüberstellung

Wie zu sehen ist, haben wir wirklich eine sehr grosse Auswahl an möglichen Lösungsvorschlägen erarbeitet. Bei so einer grossen Auswahl war es ziemlich schwierig diese gegeneinander abzuwägen. Wir haben auch festgestellt, dass Usability vielfach auch sehr subjektiv sein kann. So kann bei jedem Vorschlag ein negativer Aspekt gefunden werden, wenn danach gesucht wird. Es gibt gewisse Regeln ([NIELSEN] Kriterien, Kriterien von [QUESENBERRY], Regeln von [GALITZ] etc.), welche auch für die Entscheidung berücksichtigt worden sind. Eine Anwendung im praktischen Umfeld bietet jedoch solch einen grossen Spielraum, dass dies allein durch die Theorie kaum abgedeckt und eindeutig bestimmt werden kann.

Für die Gegenüberstellung waren für uns vor allem die SWOT-Analysen, sowie die Usability-Labs sehr hilfreich. Durch die SWOT-Analysen wurden wir uns den Vor- und Nachteilen besser bewusst und bei den Usability-Labs konnten wir vom Domain-KnowHow, sowie vom guten Feedback des Usability Experten profitieren.

Um eine gewisse Übersicht und auch eine Entscheidungshilfe zu erhalten, haben wir die Varianten in ein Akzeptanz/Machbarkeit Diagramm eingefügt. Die Y-Achse beschreibt die Akzeptanz (Feedback Auftraggeber, Usability-Experte, BA-Betreuer). Höhere Werte bedeuten bessere Akzeptanz. Die X-Achse beschreibt die technische Machbarkeit (Einschätzung von uns anhand Prototyp-Implementationen). Varianten mit hoher Machbarkeit sind einfacher und mit weniger Risiken umsetzbar.



Entscheidung/Kontroverse

Wie schon erwähnt und für uns auch ein grosser Lerneffekt, war die Tatsache, dass wir insgesamt zu viele Lösungsvarianten evaluiert haben. Dies liegt vor allem daran, dass wir jeweils versucht haben alle gewonnenen Erkenntnisse in die Vorschläge einfliessen zu lassen. Zudem sind etliche Ansätze Weiterentwicklungen und Verfeinerung vorheriger Ansätze. Nach dem ersten Usability Lab haben wir uns mit Stephan Huber geeinigt, den Fahrplan Ansatz weiterzuverfolgen. Bei der Umsetzung merkten wir jedoch, dass die variable Anzahl von parallelen Plänen technisch sehr schwierig umsetzbar ist. Wir mussten die gesamte Plan-History in einem Control darstellen. Wir wollten jedoch nicht pro Plan eine neue Spalte einfügen, da es sonst bei vielen Plänen zu seitlichem Scrolling käme. Auch eine dynamische Anordnung gemäss aktuellem Sichtfeld erschien uns nicht sehr benutzerfreundlich. Zusätzlich sahen wir Probleme beim Auf-/Zuklappen der Monate/Wochen, denn dadurch verschiebt sich die Anordnung zusätzlich. (Auf Monat-Ebene sind sie parallel, auf Wochen-Ebene nicht mehr etc.).

Dadurch haben wir nach einer technisch besser umsetzbaren Lösung gesucht, die zugleich auch bei einer beliebigen Anzahl von Plänen ohne horizontales Scrolling funktioniert. Daraus ist der 2D Ansatz entstanden. Dieser gruppiert die Daten nach Monaten. Vertikales Scrolling wird vermieden, indem die Pläne untereinander dargestellt werden. Leider war die Akzeptanz tiefer, denn die Gruppierung nach Monaten wurde als zu künstlicher Schnitt angesehen. Auch eine Gruppierung nach Plänen brachte keine Verbesserung und wurde als „Bug-Fix“ dargestellt.

Da uns die Akzeptanz sehr wichtig ist, hiess es „Zurück auf Start“. Wir verfeinerten mit Hilfe des Betreuers den Fahrplan-Ansatz. Daraus entstand der Simplified-Vertical-Timeline Ansatz. Bei der technischen Umsetzung haben wir einen vollständig neuen Ansatz verfolgt. Anstatt die Pläne in ein Grid oder ähnliches zu verschachteln, haben wir uns entschlossen, die Pläne in einem Canvas (Zeichenfläche) zu positionieren. So haben wir die volle Kontrolle wo und wie die Pläne auf dem Bildschirm gezeichnet werden. Dieser Ansatz erwies sich als Volltreffer und konnte in einem ersten Prototyp auch mit vielen Plänen ohne Performance-Probleme demonstriert werden. Vermutlich wären wir ohne den zusätzlichen Entscheidungsdruck gar nicht darauf gekommen, einen Canvas in Betracht zu ziehen, welcher schlussendlich zum Ziel führte.

Logische Architektur

The diagram illustrates a layered software architecture for a WPF application, organized into three main layers, each represented by a yellow box with a folder icon in the top-left corner:

- Start (WPF-Application)**: Contains a sub-component labeled **MainWindow.xaml**.
- GUI (User Control Library)**: Contains three sub-components: **GUI (XAML)**, **ViewModel**, and **Model**.
 - Dashed arrows indicate dependencies from **MainWindow.xaml** to **GUI (XAML)**, and from **GUI (XAML)** to **ViewModel**.
 - Dashed arrows indicate dependencies from **ViewModel** to **Model**.
 - A separate **Test (Unit Test Project)** box has dashed arrows pointing to both the **ViewModel** and **Model** components.
- Data (Class Library)**: Contains three sub-components: **Interfaces**, **Definitions**, and **Entities**.
 - A dashed arrow indicates a dependency from the **Model** component in the GUI layer to the **Interfaces** component in the Data layer.
 - A red dashed rectangle encloses the **Interfaces**, **Definitions**, and **Entities** components, with the label **Varian Code** placed to its right.

powered by Astah

Dient als Einstiegspunkt für unsere Anwendung. Prinzipiell besteht das Hauptfenster aus einem TabControl, welches in den Tabs das RT Summary-Control mit jeweils unterschiedlichen Patientendaten instanziiert.

GUI-Assembly

Beinhaltet das RT Summary-Control, also das Herzstück der Anwendung. Dieses ist nach dem Model-View-ViewModel kurz MVVM Pattern aufgebaut ist. Durch DataBinding der Views zu den ViewModels entsteht eine lose Kopplung zwischen Präsentation und Präsentations-Logik. Mittels Property-Changed-Callbacks können die Views über Änderungen an den ViewModels benachrichtigt werden. Dasselbe gilt für ViewModels und die Models. Dies wird jedoch nicht verwendet, da die RT Summary-Applikation eine reine „read-only“-Anwendung ist.

Views

Hier befinden sich die WPF-Controls. Durch die deklarative Definition mittels XAML kann die Abhängigkeit zwischen Markup und Code minimiert werden. Die Abhängigkeit zu den ViewModels kann direkt im XAML mittels DataBindings (OneWay, TwoWay etc.) definiert werden. Mit Triggers, Styles und DataTemplates können weitere reine darstellungsrelevante Eigenschaften festgelegt werden. Dadurch können (und das ist auch unser Ziel) wir den C#-Code in den Views (Code-Behind) auf ein Minimum reduzieren, was unter anderem ein Vorteil für die Testbarkeit ist. Zudem sind die Deklarationen im XAML viel weniger fehleranfällig als echte C#-Code-Zeilen.

ViewModels

Zwischenstück zwischen View und Model. Dieses stellt den Views die nötigen Daten zur Anzeige auf dem GUI bereit. Mittels Public Properties, sowie Commands (ICommand Interface) wird die Interaktion ermöglicht.

Models

Hier kommt unser eigenes Datenmodell zur Verwendung. Dieses enthält Business-Logik und relativ viele (Abbildungs-)Datenklassen.

Test-Assembly

Beinhaltet die Testklassen für die automatisierten Tests mittels Microsoft Unit Testing Framework. Getestet werden die ViewModels und Models (daher die Abhängigkeiten im Diagramm. Weitere Infos im Kapitel Testing).

Data-Assembly

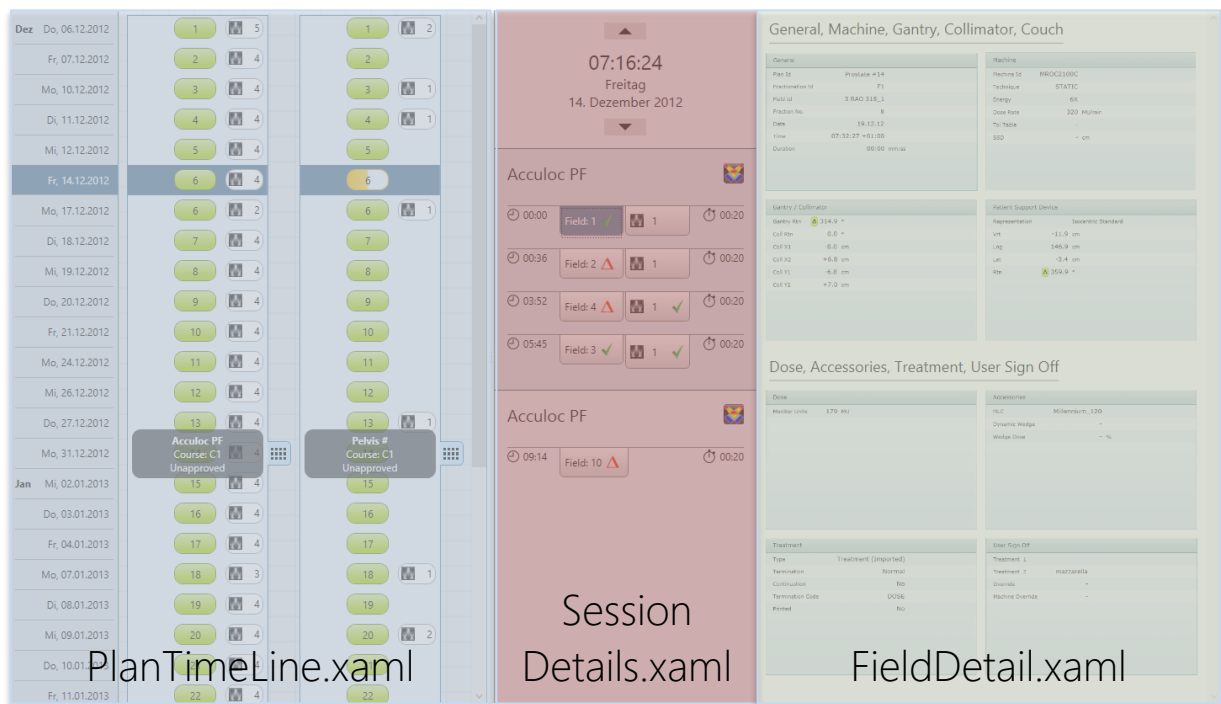
In dem Data-Assembly befinden sich die Klassen- und Interface-Definitionen für das Varian Daten-Modell. Diesen Code haben wir direkt von Varian erhalten und haben ihn auch „as is“ gelassen. Wir haben zusätzlich noch eine Methode für die Deserialisierung (in einer separaten Klasse) erstellt.

GUI-Architektur

Da der Kern unserer Lösung das Design des GUIs darstellt, möchten wir in diesem Teil genauer darauf eingehen, warum und wie wir die Oberfläche aufgebaut haben.

RTSummaryControl.xaml

Das RT Summary-Control ist das Hauptfenster, welches vom Kunden in die bestehende Anwendung integriert wird. Sie vereint die einzelnen Views und steuert den Screen-Flow (wann werden die Session-Details, wann die Bildaufnahmen-Details angezeigt/ausgeblendet etc.). Zwischen den Controls gibt es einen Grid-Splitter um die Grösse pro Control zu regeln.



Resources/Styles.xaml

Da WPF wiederverwendbares Styling erlaubt, haben wir die Styles (z.B. für den Fraction-Button) zentral in diesem File abgelegt.

Resources/DataTemplates.xaml

WPF erlaubt es die Darstellung für ViewModels (welche in ContentPresenter verwendet werden) zu definieren. Dadurch wird sichergestellt, dass z.B. ein Plan immer gleich dargestellt wird. Wir haben alle DataTemplates zentral in diesem File abgelegt. Somit fällt es leicht, Änderungen betreffend der reinen Darstellung durchzuführen.

PlanTimeLine.xaml

In diesem Control befindet sich die Darstellung der Pläne in der Zeitachse. Die Darstellung der Pläne geschieht mittels einem WPF Canvas (Zeichenfläche). Wir haben uns für einen Canvas entschieden, da für die Darstellung von parallelen Plänen andere Varianten wie z.B. Grid oder ItemsControl zu keinem Erfolg führten. Das ItemsControl eignet sich nicht für Zwei-Achsen-Darstellungen und Grids eignen sich weniger gut für grosse und dynamische Daten. Für unseren Canvas haben wir einen Algorithmus entwickelt, der die Pläne anhand des Start-Datums positioniert und darauf achtet, dass parallele Pläne einen horizontalen Offset bekommen. Die Höhe wird anhand der Anzahl Tage einmalig im Voraus nach dem Laden eines Patienten errechnet. Zusätzlich haben wir es geschafft, die Positionierungs-Logik ins ViewModel anstatt in den Code-Behind zu integrieren. Da ein WPF-Canvas normalerweise eine fixe Breite/Höhe hat, mussten wir uns mit einem Scrollable-Canvas behelfen, welcher die nötige Breite/Höhe anhand des Inhalts bestimmt. So wird die Scrollbar auch nur dargestellt, falls der Inhalt dies verlangt. Den Code dieses Canvas haben wir im Internet gefunden und auch als Fremd-Code deklariert (siehe Kapitel Deklaration Fremd-Code).

Für die Beschriftung der Tage befindet sich ein normales ItemsControl mit den relevanten Tagen auf der linken Seite. Bei diesem haben wir die Scroll-Möglichkeit ausgestellt und die Scroll-Position wird mit der Scroll-Position des Canvas synchronisiert. So entsteht der Eindruck, dass die Beschriftung im gleichen Control wie die Pläne ist, wobei diese in Wirklichkeit zwei nahtlos zusammenspielende Controls sind. Die Beschriftung der Tage kennt drei Zustände: sichtbar, aufgeklappt und versteckt. Bei jedem Plan wird jeweils der erste und letzte Tag sichtbar gemacht. Damit ist der Zeit-Bereich des Plans gut sichtbar. Aufgeklappte Tage benötigten den meisten Platz und sind dann gesetzt, wenn die betreffenden Tage beim Plan aufgeklappt worden sind. Da es auch möglich ist, dass bei einem Plan nur gewisse Tage angezeigt werden, haben wir diese Eigenschaft pro Tag festgelegt. Jede Fraction hat eine Referenz auf den Tag und weiss somit ob die Fraction-Nummer angezeigt werden soll oder nicht.

Die Höhe eines Planes definiert sich aus der Summe der Höhen der Tage. Das Konzept ist, dass die Tage je nach Zustand eine andere Grösse besitzen. Diese Grössen haben wir für die Darstellung auf einem 24" Monitor optimiert. Sie können jedoch auch in den Applikations-Properties geändert werden. Da auch bei versteckten Tagen eine gewisse Grösse gesetzt ist, stimmen auch die Grössenverhältnisse bei zugeklappten Plänen mit unterschiedlicher Anzahl von Behandlungstagen.

Beim Aufklappen eines Planes werden alle darin enthaltenen Tage auf den Zustand aufgeklappt gesetzt. Da Nachbarpläne auf die gleichen Tage referenzieren, können bei diesen automatisch über eine Property-Changed-Notification die entsprechenden Tage/Fractions angezeigt werden. Ein Plan kennt dabei drei Zustände: zugeklappt, teil-aufgeklappt (durch einen parallelen Nachbarplan) und vollständig-aufgeklappt.

SessionDetails.xaml

Dieses Control ist zuständig für die Anzeige der Session-Details. Diese enthalten Informationen über den Behandlungs-Startzeitpunkt, sowie die einzelnen Behandlungs-Elemente. Die Behandlungs-Elemente werden in einem einfachen ItemsControl dargestellt, bei welchem zusätzlich die Zeilen abwechselnd eingefärbt werden. Die Darstellung wird mittels DataTemplate definiert.

FieldDetail.xaml und ImageDetail.xaml

View für die Anzeige der Bildaufnahmen bzw. Feld-Parameter während der Behandlung. Da wir momentan vom Model aus keine Bildaufnahmen erhalten, ist diese View vor allem für die spätere Weiterentwicklung wichtig.

Deklaration Fremd-Code

Folgende Code-Files haben wir in unserer Arbeit von unten genannter Quelle verwendet.

FILE	QUELLE	ZWECK
Data/DataSource/VModel.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/EExceptionType.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/ElmageEnergy.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/ElmageSource.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/ElmageType.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/EPlanStatus.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Definitions/EPlanType.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VCourse.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VException.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VField.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VImage.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VImageOnlyEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VPatient.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VPlan.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VPlanTreatmentEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VScheduledEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VScheduledImage.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VScheduledPlan.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VScheduledSession.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VTreatmentRecord.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Entities/VTreatmentSession.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/ICourse.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IException.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IField.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IImage.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IImageOnlyEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IModel.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IPatient.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IPlan.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IPlanTreatmentEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IScheduledEvent.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IScheduledImage.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/IScheduledPlan.cs	Stephan Huber, Varian	Varian Daten-Modell

FILE	QUELLE	ZWECK
Data/Interfaces/IScheduledSession.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/ITreatmentRecord.cs	Stephan Huber, Varian	Varian Daten-Modell
Data/Interfaces/ITreatmentSession.cs	Stephan Huber, Varian	Varian Daten-Modell
GUI/Utils/DelegateCommand.cs	http://wpftutorial.net/DelegateCommand.html	Wiederverwendbarer WPF Command
GUI/Utils/ScrollableCanvas.cs	http://www.java2s.com/Tutorial/CSharp/0470_Windows-Presentation-Foundation/CreateaScrollableCanvasControl.htm	Canvas mit variabler Grösse je nach Inhalt.

Datenmodell

Vorgabe

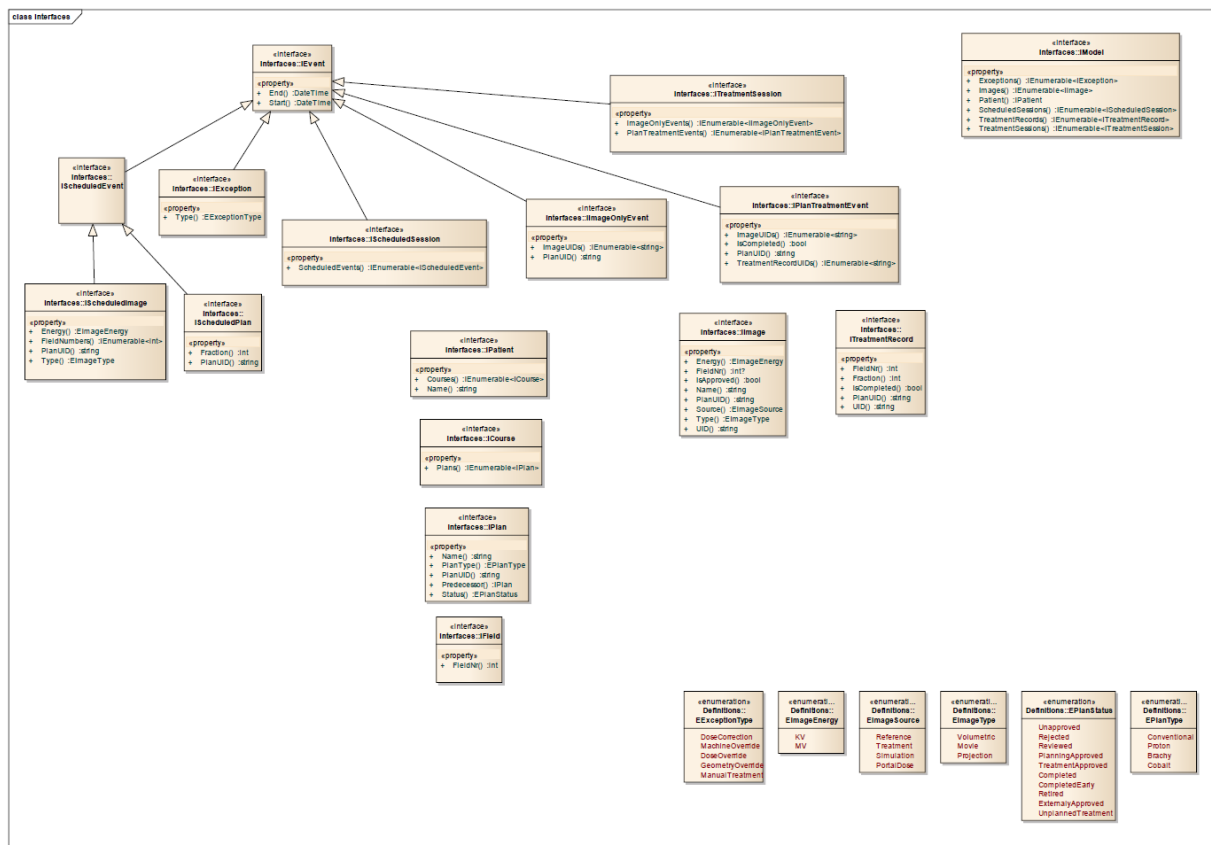


Abbildung 30: Domänenmodell-Ausgangslage

Die Daten lagen uns gemäss oben dargestellter Schnittstelle (Abbildung 30) im XML-Format vor. Diese Abbildung haben wir direkt von Varian erhalten. Leider ist diese Abbildung noch unvollständig, da die Beziehungen zwischen den einzelnen Klassen nicht eingezeichnet sind.

Nach genauer Analyse konnten wir das Domänenmodell folgendermassen aufzeichnen (Abbildung 31). Bei der Abbildung haben wir nur die wichtigsten Attribute aufgenommen, da uns vor allem die Beziehungen der Klassen untereinander interessierten. Hier fällt auf, dass die Daten doch scheinbar sehr komplex verschachtelt sind. Es gibt sehr viele „*“ zu „*“-Beziehungen was zu aufwändigerer Programmierung führen kann. Zudem sind Informationen, die auf Top-Level-Ebene benötigt werden (z.B. Bildaufnahmen pro Plan) nur sehr aufwendig über mehrere Stufen (TreatmentSession-

Architektur



42

Designentscheid „Own Model on Top“

Da wir dieses Datenmodell unserer Meinung nach nur mit sehr viel Aufwand effizient benutzen könnten, haben wir uns für ein eigenes Datenmodell entschieden (Abbildung 32). Da wir die Daten-Grundlage nicht ändern können, wird unser Modell als Aufsatz auf das bestehende verwendet (daher die Bezeichnung „on-top“). Die Daten aus dem Varian-Model werden in unser Model übersetzt und dann als Datenquelle für die ViewModels verwendet. Wir sind der Meinung, dass mit dieser zusätzlichen Schicht die spätere Verwendung in den ViewModels stark vereinfacht wird.

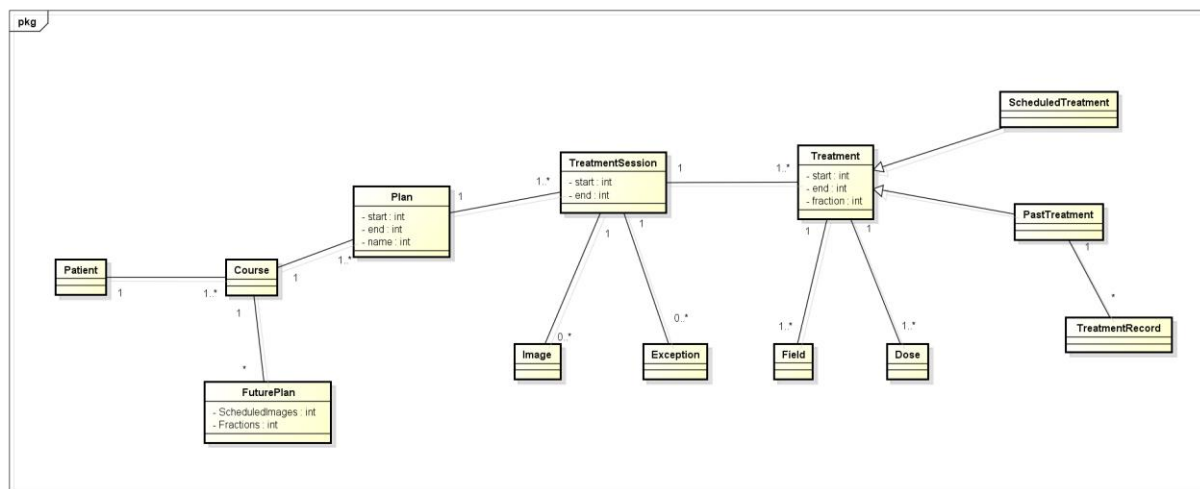


Abbildung 32: Eigenes Modell

Wir haben berücksichtigt, dass es im Varian-Model auch Pläne gibt die gar nicht zeitlich dargestellt werden können, da für diese noch kein Datum definiert ist. Im Varian-Model werden diese in das bestehende Modell hineingedrückt und haben dann im Datumsfeld den Sentinel-Wert „1.1.0001“. Bei uns werden diese speziellen Pläne separat in der Klasse FuturePlan gehalten. So können diese im GUI sehr einfach getrennt dargestellt werden. In unserem Modell werden jeweils auch alle Objekte im Objektmodell referenziert und nicht über IDs (bzw. UUIDs), wie bis anhin. Der Prozess zusätzlich ein eigenes Modell zu erstellen hat uns zwar einige Zeit gekostet, jedoch erlaubt dieses eine erhebliche Vereinfachung für die Verwendung in den ViewModels.

Konsequenzen bezüglich Update-Fähigkeit

Die RT Summary-Applikation ist eine reine „read-only“-Applikation. In der Medizin ist dies vielfach der Fall, da es sich um vergangene Daten handelt und diese sich nicht verändern. Daher konnten wir mit gutem Gewissen ein eigenes Modell über dem bestehenden Modell legen, ohne uns gross um die Rückabbildung zu konzentrieren. Wären Rückabbildungen notwendig, müsste dies zusätzlich implementiert werden. (Projektion von unserem Modell zu Varian-Datenmodell). Dies gestaltet sich oft einfach (Plan wrappt ein VPlan-Objekt, TreatmentRecord wrappt ein VTreatmentRecord-Objekt etc.). Gewisse Aspekte benötigen jedoch etwas höheren Aufwand z.B. Sessions.

Austauschbarkeit/Integration

Durch gute Schichtenbildung ist es später auch möglich die Daten anstatt über ein XML-File, direkt über eine Datenbank zu beziehen. Dazu muss nur die Data-Assembly angepasst werden um dort die Daten aus der Datenbank zu laden. Für eine erste Integration haben wir uns auf eine Schnittstelle mit (File-)Stream Übergabe geeinigt. Ein erster Test beim Projektpartner war erfolgreich, denn er konnte unser Control im bestehenden Informationssystem einbinden und mit den echten Daten befüllen.

Testing

Unit-Tests

Da wir den MVVM-Ansatz für unsere Lösung gewählt haben, ist trotz grossem GUI-Anteil eine hohe Testabdeckung mittels Unit-Tests möglich, da die GUI-Logik entkoppelt in den ViewModels liegt. Vollständigkeitshalber muss erwähnt werden, dass auch im reinen XAML-Markup ein gewisser Teil Programm-Logik vorhanden ist. Dies vor allem in Form von Triggers und DataBindings enthalten, welche auch Fehler enthalten könnten. Solch ein fehlerhaftes Verhalten kann jedoch ziemlich schnell bei Systemtests gefunden werden, da sich die Oberfläche dadurch nicht wie erwartet verhält. Zudem hatten wir während der Entwicklung auch ein Auge auf das Output-Window von Visual Studio geworfen, denn dort werden fehlerhafte Bindings während der Ausführung angezeigt.

In der Unit-Test Assembly haben wir insgesamt 86 Testmethoden definiert. In diesen Methoden testen wir vor allem unser eigenes Modell, sowie die ViewModels. Mit diesen Tests erreichen wir eine sehr gute Testabdeckung von 94,74% (Abbildung 33). Solch eine hohe Zahl ist jedoch trotzdem mit Vorsicht zu geniessen, denn die Testabdeckung macht nur eine Aussage über die Abdeckung der statischen Programmpfade und nicht über die dynamische Wertekombination der Variablen, die das Verhalten steuern.

Hierarchy ▲	Covered (Blocks)	Not Covered (% Blocks)	Covered (% Blocks)
stefanleimgruber_STEFAN 2014-06-08 00_08_37.coverage	3281	3,87%	96,13%
vms.rtm.rtsummary.data.dll	0	100,00%	0,00%
vms.rtm.rtsummary.gui.dll	1764	5,26%	94,74%
{ } VMS.RTM.RTSummary.GUI	0	100,00%	0,00%
{ } VMS.RTM.RTSummary.GUI.Models	503	0,00%	100,00%
{ } VMS.RTM.RTSummary.GUI.Properties	5	0,00%	100,00%
{ } VMS.RTM.RTSummary.GUI.Utils	34	41,38%	58,62%
{ } VMS.RTM.RTSummary.GUI.ViewModels	1222	2,24%	97,76%
{ } VMS.RTM.RTSummary.GUI.Views	0	100,00%	0,00%
vms.rtm.rtsummary.test.dll	1517	1,75%	98,25%

Abbildung 33: "Code Coverage Results" für unsere Solution

Da wir das Datenmodell mit Varian definiert haben, haben wir für unsere Unit-Tests eine separate Klasse erstellt, welche einen Test-Patienten mit allen benötigten Kombinationen (parallele Pläne, nicht vollumfänglich durchgeführte Behandlungen, geplante Behandlungen etc.) erstellt. Damit konnten wir alle Programmpfade testen.

System-Tests

Für die System-Tests haben wir insgesamt vier Patienten verwendet. Zwei Patienten wurden von uns erstellt, wobei einer auch für die Unit-Tests verwendet wurde (TestData.xml). Die anderen zwei waren von Stephan Huber zur Verfügung gestellte Test-Patienten (Brooks.xml und FranzKafka.xml). Soweit wir wissen, sind diese zwei Patienten jedoch synthetische oder stark anonymisierte Patienten zu Demo-Zwecken. Diese können deswegen von uns problemlos zu Test- und Demonstrations-Zwecken eingesetzt werden. In der nachfolgenden Tabelle sind die System-Tests ersichtlich. Diese wurden manuell durchgeführt. (Spalte E = Ergebnis)

NR.	TESTABLAUF	ERWARTETES RESULTAT	E
#1	Planübersicht öffnen	Pläne werden dargestellt inkl. paralleler Pläne. Pläne sind jeweils mit dem Start- und Enddatum beschriftet, sowie Plan-Name, Kurs, Plan-Status, Fortschritt und die Anzahl der Bildaufnahmen sowie die Anzahl der Ausnahmen (falls vorhanden)	✓
#2	Scrolling in der Planübersicht	Pläne verschieben sich entsprechend. Spätere Pläne werden sichtbar. Datumsbereich verschiebt sich entsprechend. Scrolling ohne Verzögerung möglich.	✓
#3	Window-Resizing	Scrollbars werden je nach Bedarf ein- und ausgeblendet.	✓
#4	Plan aufklappen	Plan wird aufgeklappt und es werden die einzelnen Fractions angezeigt. Der Datumsbereich zeigt das Datum pro Fraction an. Bei parallelen Plänen werden die gleichen Tage aufgeklappt, der Rest bleibt verborgen, sofern noch zusätzliche Tage vorhanden sind.	✓
#5	Vergangene Behandlung (grün ausgefüllt) auswählen	Session-Details für diesen Tag werden angezeigt.	✓
#6	Nicht vollumfänglich durchgeführte Behandlung (halb orange ausgefüllt) auswählen	Session-Details für diesen Tag werden angezeigt.	✓
#7	Geplante Behandlung (blau ausgefüllt) auswählen	Kann nicht ausgewählt werden.	✓
#8	Session-Details anzeigen	Gruppiert nach Plan und Zeit. Die relative Zeit und Dauer einer Behandlung wird jeweils angezeigt. Zusätzlich wird bei der relativen Zeit im Tooltip die absolute Zeit dargestellt. Tag wird in der Planübersicht durchgehend (über alle Pläne) blau markiert.	✓
#9	Zum nächsten/vorherigen Tag springen	Mittels Pfeil-Buttons kann in den Session-Details zwischen den Tagen navigiert werden. Am Ende und beim ersten Tag kann nur in die andere Richtung navigiert werden.	✓
#10	Feld auswählen (Session-Details)	Feld wird selektiert. Die letzte Feld- bzw. Bildaufnahme-Selektion wird deselektiert.	✓
#11	Bildaufnahme auswählen (Session-Details)	Bildaufnahme wird selektiert. Die letzte Feld- bzw. Bildaufnahme-Selektion wird deselektiert.	✓

Resultat

Nach einer sehr langen, jedoch erfolgreichen Studie zu Beginn der Arbeit, hatten wir trotz überschreiten der geplanten Zeit, eine sehr umfangreiche Applikation erstellt. Das Ziel der Studie, bzw. der Analysephase war, dass wir uns mit den Daten auseinandersetzen und neue Möglichkeiten aufzeigen, wie diese Daten dargestellt werden können. Wir hatten die Entscheidungen nicht aus dem Bauchgefühl heraus getroffen, sondern zusammen mit unserem Projektpartner und Betreuer erarbeitet. Wir wendeten unter anderem das Card-Sorting an, bei welchem wir herausfinden wollten, wie die Prioritäten der anzuzeigenden Daten sind. Daran orientierten wir uns während der Erstellung der möglichen Lösungsvarianten. Uns war wichtig, dass sich die Benutzer aus allen Berufsgattungen schnell zu Recht finden und auch ihre Informationen auf einem Blick erhalten. Des Weiteren wollten wir die Möglichkeit anbieten, dass Informationen, welche momentan nicht relevant sind, zugeklappt werden können.

Auf das Resultat sind wir sehr stolz und es lässt sich zeigen, denn es wurden (wie in den folgenden Kapiteln aufgezeigt) sehr viele Features, welche vom Projektpartner gewünscht/definiert worden sind, umgesetzt.

Auswertung Analysephase

Um sich ein Bild über die Analysephase verschaffen zu können, haben wir eine Auswertung zusammengestellt.

ERGEBNIS	ANZAHL	BESCHREIBUNG
Domänenmodelle	2	Bei der Domänenanalyse mussten wir die Ausgangslage, welche wir vom Projektpartner erhalten haben nochmals überdenken und ein eigenes Domänenmodell erstellen um auch die Zusammenhänge klar zu sehen. Des Weiteren erstellten wir in der Implementationsphase ein weiteres Domänenmodell, welches die Schnittstelle für unsere Implementierung vereinfachte.
Lösungsvarianten	11	Insgesamt wurden elf Lösungsvarianten erarbeitet.
Wireframes	45	Zu jeder Lösungsvariante erstellten wir im Durchschnitt 4-5 Wireframes um unsere Ideen auch visuell darzustellen und mit dem Projektpartner und Betreuer besprechen zu können.
Implementierte Wireframes	4	Von allen Lösungsvarianten, welche wir weiterverfolgt haben, haben wir C# Prototypen erstellt um auch die Eignung der technische Umsetzung (POC) sicherzustellen.
Papierprototypen	2	Zwei von den elf Lösungsvarianten hatten wir auch als Papierprototypen erstellt und diese dann von Hand simuliert.
PowerPoint-Prototypen	12	Bei allen Ansätzen hatten wir mindestens einen PowerPoint-Prototyp erstellt. Dabei wurden die Wireframes mit klickbaren Schaltflächen versehen.
Aufwand	182.5h	Zusammengefasst machte die Analysephase bei uns ca. 25% des gesamten Projektaufwandes (ohne Aufwand für Prototypen-Implementierung) aus.

Vergleich Ausgangslage / Resultat

In der folgenden Tabelle vergleichen wir die Ausgangslage mit dem Resultat.

ASPEKT	ALT	NEU
Overview	Schlechte Übersicht. Die Oberfläche ist mit Informationen überladen.	Sehr gute Top-Level Übersicht über die Pläne. Keine Überladung; nur die relevanten Informationen werden angezeigt (bzw. zusammengefasst).
Details-on-demand	Keine Möglichkeit die Granularität pro Plan festzulegen.	Granularität der anzeigenden Informationen kann pro Plan Level bestimmt werden (Plan auf-/zuklappen).
Date-Range	Schwierig ersichtlich über welchen Zeitraum die Behandlungen stattgefunden haben. Unnötiger Freiraum zwischen den Behandlungen-Zeiträumen. Der Startpunkt ist immer beim aktuellen Datum, auch wenn die letzte Behandlung bereits Jahre zurückliegt. Fast endloses Scrolling bei Patienten-Daten über mehrere Jahre nötig.	Es wird nur der Datumsbereich von aktiven Behandlungen angezeigt. So ist auf einen Blick ersichtlich in welchem Zeitraum der Patient in Behandlung war. Unnötige Zeiträume in welchen keine Behandlungen stattgefunden haben sind ausgeblendet und verschwenden keinen Platz. Übersichtlich Darstellung auch bei Daten mit einem Zeitraum von mehreren Jahren.
Course-View	Nur jeweils ein Kurs sichtbar. Kurs-Wechsel nötig. Kein Gesamtbild des Patienten über alle Kurse möglich.	Kein Wechsel mehr nötig. Es werden alle Pläne aus allen Kursen in einer Darstellung vereint.
Initial Load	Spürbar aber ertragbar	Kaum spürbar
User-Interaction	Spürbare Verzögerung beim Scrolling. Zusätzliche Wartezeit bei der Anzeige der Sitzungs-Details.	Weder beim Scrolling noch bei der Anzeige der Sitzungs-Details eine Verzögerung spürbar.
Image-View	Sehr kleine Darstellung	Grosse Darstellung, da mehr Platz für Bilder im Hochformat vorhanden.

Umgesetzte Features

Hier werden unter anderem die in der Anforderungsanalyse definierten und umgesetzten Features aufgelistet und jeweils mit einem Screenshot hinterlegt. Des Weiteren sind die Anforderungen, welche während des Verlaufs des Projektes entstanden sind, aufgeführt.

Zu Beginn definiert

FEATURE

SCREENSHOT

Course Timeline

Da sich eine Gruppierung nach Kursen als nicht sinnvoll herausgestellt hat, wurden die Pläne jeweils mit dem Kurs angeschrieben.

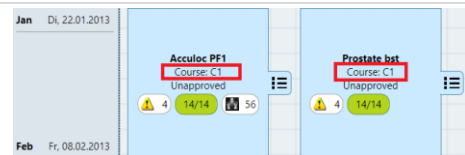


Abbildung 34: Kurs Beschriftungen

Involved Plans

Die Pläne sind als Gruppierungseinheit in der Top-Level-Anzeige ersichtlich. Sie werden automatisch angeordnet und in der Zeitachse dargestellt.

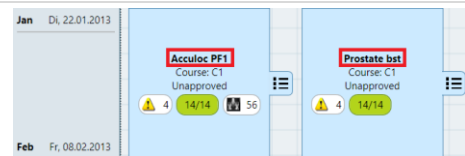


Abbildung 35: Plan Beschriftungen/Gruppierung

Past Treatments

Bei den vergangenen Behandlungen gibt es grundsätzlich zwei Arten der Darstellung. Diejenigen Behandlungen, welche vollumfänglich durchgeführt wurden (siehe Screenshot; grün markiert) und diejenigen, welche nur teilweise durchgeführt wurden siehe unten Partial Treatments.



Abbildung 36: Vergangene Behandlungen

Partial Treatments

Bei der Anzeige der nicht vollumfänglich durchgeführten Behandlungen wird die Behandlung nur halb ausgefüllt und gelb dargestellt. Dadurch ist schnell und auf einen Blick ersichtlich, bei welchen Behandlungen evtl. reagiert werden muss.



Abbildung 37: Partielle/Fehlerhafte Behandlungen

Future Treatments

Zukünftige Behandlungen sind durch ihre Farbe erkennbar. Diese Behandlungen sind geplant und besitzen ein Datum und eine Uhrzeit, was der Unterschied zu den zukünftigen Plänen (siehe weiter unten Futur Plans) ist.

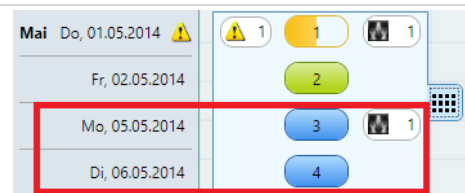


Abbildung 38: Zukünftige Behandlungen

Field Parameters

Bei den Feld-Parametern handelt es sich um die Detailinformationen nach der Selektion eines Feldes. Diese mussten wir mocken (vortäuschen), da wir dazu keine Testdaten erhalten haben.

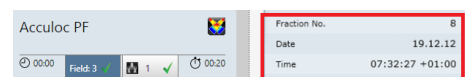


Abbildung 39: Detailinformationen zu Feldern

FEATURE

Imaging Activities

Falls Bildaufnahmen in einem Plan vorkommen, werden diese bereits Top-Level angezeigt (Abbildung 40). Wenn der jeweilige Plan geöffnet wird, werden die Bilder jeweils an diesem Tag angezeigt (Abbildung 41), an welchem sie aufgenommen worden sind. Bei der Detailansicht der Bildaufnahmen erhält der Benutzer die Information, ob ein Bild approved (angenommen) worden ist oder nicht (Abbildung 42 & Abbildung 43). Die Bildaufnahmen enthalten in einem Tooltip noch zusätzliche Informationen. (Abbildung 44)

Der Prozess der Bildaufnahmen-Genehmigung ist nicht implementiert und als weitere Idee im Kapitel Weitere Ideen/Open-Issues beschrieben.

SCREENSHOT

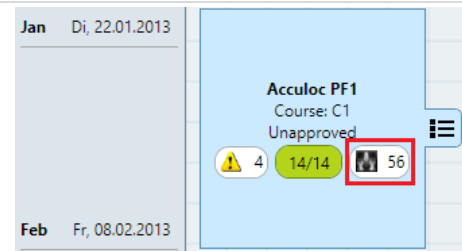


Abbildung 40: Bildaufnahmen Top-Level

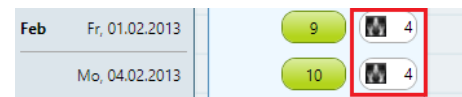


Abbildung 41: Bildaufnahmen Tagesansicht



Abbildung 42: Bildaufnahmen Detailansicht

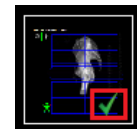


Abbildung 43: Bildaufnahmen Detailansicht – Approval

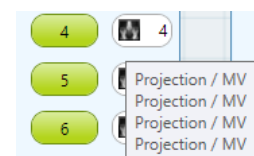


Abbildung 44: Bildaufnahmen Tooltip

Exceptions / Overrides

Die Ausnahmen, bzw. Overrides (Änderungen von Werten) werden auf der Top-Level-Anzeige eines Plans angezeigt (Abbildung 45), sobald mind. eine Ausnahme / Änderung vorhanden ist. Diese werden dann beim Öffnen des Planes am jeweiligen Tag angezeigt (Abbildung 46). Da die Testdaten so aufgebaut sind, dass eine Ausnahmen / Änderungen nicht direkt einem Plan zugeordnet werden kann, werden die Ausnahmen / Änderungen jeweils bei beiden Plänen am jeweiligen Datum dargestellt (Abbildung 47). Die Informationen werden momentan in einem Tooltip angezeigt (Abbildung 48).

Die detaillierte Anzeige der Ausnahmen / Änderungen ist nicht implementiert und wird ebenfalls als weitere Idee im nächsten Kapitel beschrieben.



Abbildung 45: Ausnahmen/Änderungen Top-Level

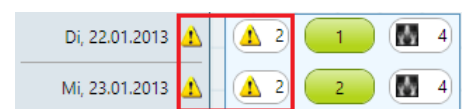


Abbildung 46: Ausnahmen/Änderungen Tagesansicht

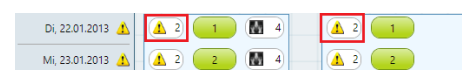


Abbildung 47: Ausnahmen/Änderungen Beschriftung

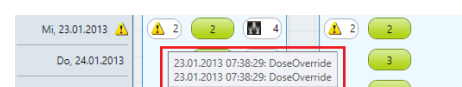


Abbildung 48: Ausnahmen/Änderungen Tooltip

Im Verlauf des Projektes definiert

Nachfolgend sind die Features aufgelistet, welche im Verlauf des Projektes dazugekommen sind.

FEATURE

SCREENSHOT

Future Plans

Nach der Analyse der Daten ist uns aufgefallen, dass gewisse geplante Pläne, bzw. Behandlungen kein Datum und Zeitangaben besitzen. Es stellte sich heraus, dass dies Pläne sind, welche geplant sind, jedoch noch keine Angaben zur zeitlichen Durchführung haben. Der Projektpartner wollte diese ebenfalls in der Zeitachse ersichtlich haben. Somit entschieden wir, dass diese am Ende als „Future Plans“ dargestellt werden (Abbildung 49).



Abbildung 49: Zukünftige Pläne ohne Datum und Zeit

Plan Status

Jeder Plan enthält einen Status. Dieser ist je nach Stand der Behandlungen unterschiedlich. Um diesen gleich zu sehen, haben wir den Status zu Beginn in der Top-Level-Anzeige dargestellt (Abbildung 50).

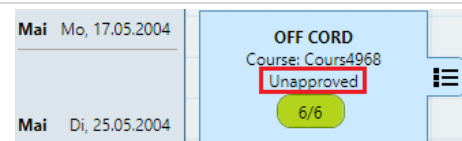


Abbildung 50: Plan-Status

Zudem wurden viele kleinere und grössere Anforderungen an die Anzeige definiert. Diese wurden teilweise bereits durch einen Screenshot dargestellt. Deshalb enthält diese Tabelle nicht bei jedem Feature einen Screenshot, sondern nur einen Verweis auf ein bereits beschriebenes Feature. Die Features, welche nicht umgesetzt oder gestrichen wurden, werden im nächsten Kapitel beschrieben. (Spalte U = Umgesetzt)

FEATURE	WIE UMGESETZT / SCREENSHOT	U
Wenn ein Plan zugeklappt ist, dann sollen Batches für Ausnahmen/Bildaufnahmen anzeigen.	Siehe Imaging activities und Exceptions/Overrides	✓
Tick bei approved Images.	Siehe Imaging activities	✓
Wochentag anzeigen z.B. FR, 16.05.2014	Siehe z.B. Abbildung 45	✓
Bildaufnahmen in Session-Details anzeigen.	Siehe Imaging activities	✓
Icons verwenden, zusätzlich zur Farbcodierung.	Auf diversen Screenshots ersichtlich	✓
Anzeigen, wenn ein Plan nur teilweise geöffnet ist.	Andere Hintergrundfarbe	✓
Monate in der Zeitachse anzeigen.	Siehe Abbildung 51	✓
Scheduled Plans (zukünftige Pläne) ohne Datum in der Timeline anzeigen.	Siehe Future Plans	✓
Minimum-Size für Plan (wenn zugeklappt/offen).	Da die Grösse von der Anzahl Fractions abhängig ist, haben wir sichergestellt, dass ein Plan ab 4 Fractions zugeklappt vollständig dargestellt werden kann.	✓



Abbildung 51: Plan rechts nur teilweise geöffnet

FEATURE

WIE UMGESETZT / SCREENSHOT

U

Bei den Detailinformationen soll das Datum, der Wochentag und die Zeit des Startes angezeigt werden.

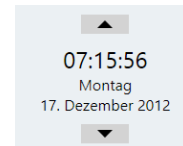


Abbildung 52: Detailinformationen Beschreibung

Bei der Selektion eines Feldes oder der Bildaufnahmen sollen diese markiert werden.

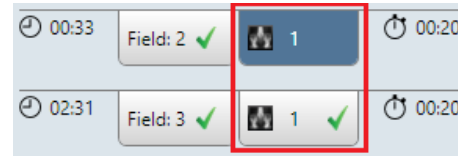


Abbildung 53: Markierung bei Selektion

Detailinformationen ausblenden, wenn Pläne zugeklappt sind. Wurde so umgesetzt.



Icons für Dauer und relative Zeit bei Detailinformationen (Stoppuhr- und Uhr-Icon).



Abbildung 54: Icons für Dauer und relative Zeit

Keyboard-Bedienung, z.B. mit Pfeile für Fractions navigieren.

Hinweis: Wurde absichtlich nicht implementiert, da davon abgeraten wird. Als Alternative haben wir Pfeile im Detailinformationen-Header eingefügt um dadurch navigieren zu können.

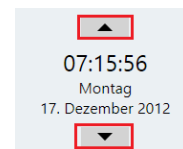


Abbildung 55: Navigation durch Tages-Details

Delta-Icon für partielle Felder in den Detailinformationen.



Abbildung 56: Delta Icon für partielle Felder

Wenn keine Details zu Feldern oder Bildaufnahmen angefordert werden, soll dies ersichtlich sein.

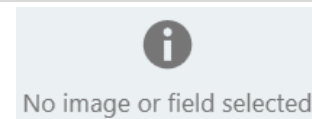


Abbildung 57: Keine Selektion Anzeige

Icons für öffnen/zuklappen eines Planes.



Abbildung 58: Icon Plan öffnen

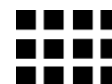


Abbildung 59: Icon Plan zuklappen

Weitere Ideen / Open Issues

In der folgenden Tabelle haben wir Ideen notiert, welche in zukünftigen Versionen noch zusätzlich umgesetzt werden könnten, welche wir jedoch aus Zeitgründen bzw. in Anbetracht des Scopes nicht realisiert haben.

IDEE	BESCHREIBUNG
Fractions-Fortschritt anzeigen	So wie der Fortschritt in der Behandlung angezeigt wird, könnten ebenfalls die Fractions teilweise ausgefüllt werden, falls diese nicht vollumfänglich durchgeführt wurden. Dies ist grundsätzlich nur bei Fractions, welche nicht vollumfänglich durchgeführt wurden sinnvoll.
Nicht vollumfänglich durchgeführte Behandlungen in der Planübersicht anzeigen	Hier könnte die Anzahl der nicht vollumfänglich durchgeführten Behandlungen in der Planübersicht dargestellt werden, damit gleich ersichtlich ist, wie viele Behandlungen nicht vollumfänglich durchgeführt wurden.
Bildaufnahmen-Genehmigung	Der Prozess der Genehmigung einer Bildaufnahme wurde nicht implementiert, da dies nicht im Scope der Bachelorarbeit war. Dies könnte jedoch in der Bildaufnahmen-Detailansicht eingebaut werden.
Ausnahmens-Anzeige	Die Ausnahmen sind momentan nicht weiter ins Detail angezeigt worden, bzw. nur in einem Tooltip. Dies könnte noch weiter verfeinert werden, dass z.B. die Informationen gleich wie bei den Felder-Details und Bilderaufnahmen-Details angezeigt werden.
Image Only Events	Es gibt die Möglichkeit „Image Only Events“ zu planen. Dies sind Events, an welchen nur Bilder gemacht werden. Dies wurde von uns ebenfalls aus Zeitgründen und dem Scope nicht implementiert. Jedoch könnten diese Informationen in der Fraction-Ansicht, d.h. wenn ein Plan geöffnet wird speziell gekennzeichnet, angezeigt werden.
Allgemein Mocking	Da wir nicht alle Daten zur Verfügung hatten, mussten wir an vielen Orten noch eigene Mockings (Vortäuschungen) einbauen um die Anzeigen zu füllen. Diese müssen natürlich für einen produktiven Einsatz mit echten Daten gefüllt werden. (siehe FieldDetail.xml und ImageDetail.xml)
Filterfunktion (nach Datum, Fraction, Dosis, etc..)	Ein weiteres Feature wäre eine Filterfunktion, mit welcher nach bestimmten Kriterien gefiltert werden können.
Suchfunktion (bestimmte Daten, Datum, etc..)	Weiter war eine Suchfunktion eine Idee von uns, dass nach gewissen Daten auch gesucht werden kann.
Einstiegspunkt für Doctor und Therapist	Ein sehr angenehmes Feature wäre noch gewesen, dass z.B. durch ein Login jeweils erkannt wird welche Berufsgattung die Applikation nutzt und somit auch der entsprechenden Startpunkt, bzw. Einstiegspunkt automatisch ausgewählt wird.

Auswertung Arbeitsaufwand

Arbeitsaufwand Total

Für die Bachelorarbeit ist ein Arbeitsaufwand von insgesamt 720 Stunden (12 ECTS = 360 Stunden * 2 Personen) vorgegeben. Wir haben einen Arbeitsaufwand von 788 Stunden geleistet. Dies entspricht einem Mehraufwand von 68 Stunden.

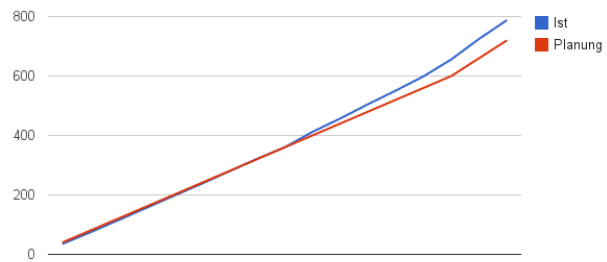


Abbildung 60: Arbeitsaufwand Total in Stunden

Arbeitsaufwand pro Aktivität

Die geleisteten Stunden, aufgeteilt nach Aktivität. Die Implementation ist mit 299 Stunden der grösste Posten, unter welche auch die Entwicklung der Prototypen fällt. Die Verfassung des technischen Berichtes ist zusammen mit der Analyse der zweite grosse Posten und zugleich ein wichtiger Aspekt der Bachelorarbeit. Insgesamt sind wir mit der Aufteilung pro Aktivität zufrieden und haben keine grossen Überraschungen entdeckt.

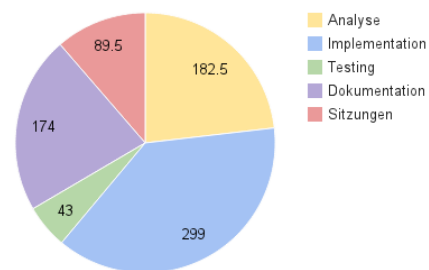


Abbildung 61: Arbeitsaufwand pro Aktivität in Stunden

Glossar

BEGRIFF	ERLÄUTERUNG
Ausnahmen/Änderungen	Fehler während der Behandlung oder Änderungen an den Behandlungswerten
Behandlung	Medizinische Untersuchung mit dem Fokus auf Krebstherapie
Bildaufnahme	Diagnostische Bildaufnahme im Zusammenhang der Behandlung
Feld	Bestrahlungs-Feld im 3-dimensionalen Raum
Fraction	Informationseinheit über einen Plan und dessen Felder
Kurs	Gruppierungseinheit für Pläne mit dem gleichen therapeutischen Ziel
Mocking	Vortäuschung von Daten während der Entwicklung (wenn die echten Daten nicht vorliegen)
Onkologie	Wissenschaft welche sich mit der Krebsbehandlung befasst
Plan	Teil eines therapeutischen Zieles
POC	Proof of Concept, Machbarkeitsstudie durch Entwicklung eines Prototyps. [POC]
RUP	Rational Unified Process, Projektmanagement-Methode [LARMAN]
Strahlentherapie	Medizinisches Fachgebiet für Behandlungen mittels Strahlung
SWOT-Analyse	Wirtschaftliches Instrument zur Positionsbestimmung von Strategien
Visual Studio Online	Zentrale Ablage von Microsoft für Projektdaten in der Cloud. Mit Tools für Projektmanagement, Source Control und Build-Service. [VSONLINE]
Wireframe	Technischer Entwurf für grafische Benutzeroberflächen

Literaturverzeichnis

ABK.	LITERATUR
[WSB]	Wikipedia: Strahlenbehandlung. (30. Mai 2014). Von http://de.wikipedia.org/wiki/Strahlentherapie abgerufen
[WONK]	Wikipedia: Onkologie. (30. Mai 2014). Von http://de.wikipedia.org/wiki/Onkologie abgerufen
[GALITZ]	Galitz, W. O. (2007). The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. Indianapolis: Wiley.
[NIELSEN]	Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.
[QUESENBERY]	WhitneyQuesenbery. (kein Datum). Von http://www.wqusability.com/handouts/dimensions-of-usability.pdf abgerufen
[POC]	Wikipedia: Proof of Concept. (kein Datum). Von http://de.wikipedia.org/wiki/Proof_of_Concept abgerufen
[LARMAN]	Larman, C. (2005). <i>UML 2 und Patterns angewendet</i> . Verlagsgruppe Hüthig Jehle Rehm GmbH.
[VSONLINE]	Wikipedia: Visual Studio Online. (kein Datum). Von http://en.wikipedia.org/wiki/Visual_Studio_Online abgerufen

Abbildungsverzeichnis

ABBILDUNG 1: CARD-SORTING AUSWERTUNG	20
ABBILDUNG 2: PLAN-/BEHANDLUNGSÜBERSICHT	23
ABBILDUNG 3: BEHANDLUNGSMITTELSINFORMATIONEN	23
ABBILDUNG 4: FELD-DETAILS	23
ABBILDUNG 5: AUSWAHL EINES Kurses -> ANZEIGE DER PLÄNE	24
ABBILDUNG 6: ZOOM-IN FÜR MEHR INFORMATIONEN	24
ABBILDUNG 7: VERTIKALE ZEITACHSE MIT DETAILINFORMATIONEN	25
ABBILDUNG 8: TAGESAUSWAHL MIT MEHR INFORMATIONEN	25
ABBILDUNG 9: DETAIL-INFORMATIONEN ZUR BEHANDLUNG	25
ABBILDUNG 10: ÜBERSICHT ÜBER KURSE EINES PATIENTEN	26
ABBILDUNG 11: TAGES-ANSICHT DER BEHANDLUNGEN	26
ABBILDUNG 12: DETAIL-INFORMATIONEN EINER BEHANDLUNG	26
ABBILDUNG 13: KURS-ÜBERSICHT DES PATIENTEN	27
ABBILDUNG 14: ZOOM-IN FÜR MEHR INFORMATIONEN EINER BEHANDLUNG	27
ABBILDUNG 15: BILDAUFNAHMEN-GENEHMIGUNG	27
ABBILDUNG 16: DETAIL-INFORMATIONEN EINER BEHANDLUNG	27
ABBILDUNG 17: MONAT/WOCHE AUFGEKLAFFT	28
ABBILDUNG 18: ZUGEKLAFFT ANSICHT	28
ABBILDUNG 19: MONAT/WOCHE AUFGEKLAFFT	29
ABBILDUNG 20: TAG AUSGEWÄHLT	29
ABBILDUNG 21: KURS-ÜBERSICHT EINES PATIENTEN	30
ABBILDUNG 22: HIERARCHIE DER DATEN ALS BAUM DARGESTELLT	30
ABBILDUNG 23: TIEFER IN DIE HIERARCHIE FÜR MEHR INFORMATIONEN	30
ABBILDUNG 24: BEHANDLUNGS-DETAILS	30
ABBILDUNG 25: DETAIL-ANSICHT EINER BEHANDLUNG; GRUPPIERUNG NACH MONAT	31
ABBILDUNG 26: INFORMATIONEN EINER BEHANDLUNG	31
ABBILDUNG 27: AUFGEKLAFFT ANSICHT EINES PLANS MIT DER TAGESANSICHT	32
ABBILDUNG 28: AUFGEKLAFFTER PLAN MIT AUFGEKLAFFTEN FRACTIONS DES PARALLELEN PLANS	33
ABBILDUNG 29: PACKAGE-DIAGRAMM DER LOGISCHEN ARCHITEKTUR	36
ABBILDUNG 30: DOMÄNENMODELL-AUSGANGSLAGE	41
ABBILDUNG 31: DOMÄNENMODELL NACH DER ANALYSE DER VORGABE	42
ABBILDUNG 32: EIGENES MODELL	43
ABBILDUNG 33: "CODE COVERAGE RESULTS" FÜR UNSERE SOLUTION	44
ABBILDUNG 34: KURS BESCHRIFTUNGEN	48
ABBILDUNG 35: PLAN BESCHRIFTUNGEN/GRUPPIERUNG	48
ABBILDUNG 36: VERGANGENE BEHANDLUNGEN	48
ABBILDUNG 37: PARTIELLE/FEHLERHAFT BEHANDLUNGEN	48
ABBILDUNG 38: ZUKÜNFTIGE BEHANDLUNGEN	48
ABBILDUNG 39: DETAILINFORMATIONEN ZU FELDERN	48
ABBILDUNG 40: BILDAUFNAHMEN TOP-LEVEL	49
ABBILDUNG 41: BILDAUFNAHMEN TAGESANSICHT	49
ABBILDUNG 42: BILDAUFNAHMEN DETAILANSICHT	49
ABBILDUNG 43: BILDAUFNAHMEN DETAILANSICHT – APPROVMENT	49
ABBILDUNG 44: BILDAUFNAHMEN TOOL TIP	49
ABBILDUNG 45: AUSNAHMEN/ÄNDERUNGEN TOP-LEVEL	49
ABBILDUNG 46: AUSNAHMEN/ÄNDERUNGEN TAGESANSICHT	49
ABBILDUNG 47: AUSNAHMEN/ÄNDERUNGEN BESCHRIFTUNG	49
ABBILDUNG 48: AUSNAHMEN/ÄNDERUNGEN TOOL TIP	49
ABBILDUNG 49: ZUKÜNFTIGE PLÄNE OHNE DATUM UND ZEIT	50
ABBILDUNG 50: PLAN-STATUS	50
ABBILDUNG 51: PLAN RECHTS NUR TEILWEISE GEÖFFNET	50
ABBILDUNG 52: DETAILINFORMATIONEN BESCHREIBUNG	51
ABBILDUNG 53: MARKIERUNG BEI SELEKTION	51
ABBILDUNG 54: ICONS FÜR DAUER UND RELATIVE ZEIT	51
ABBILDUNG 55: NAVIGATION DURCH TAGES-DETAILS	51
ABBILDUNG 56: DELTA ICON FÜR PARTIELLE FELDER	51
ABBILDUNG 57: KEINE SELEKTION ANZEIGE	51
ABBILDUNG 58: ICON PLAN ÖFFNEN	51
ABBILDUNG 59: ICON PLAN ZUKLAPPEN	51
ABBILDUNG 60: ARBEITSAUFWAND TOTAL IN STUNDEN	53
ABBILDUNG 61: ARBEITSAUFWAND PRO AKTIVITÄT IN STUNDEN	53