

Lasten- und Pflichtenheft Projekt HORUS

*3D Visualisierung von PET/CT Untersuchungen **

der Bachelor Thesis 2008
(18. Februar bis 13. Juni)

Alain Muttscheller, mutta@bfh.ch
Peter von Niederhäusern, vonnnp@bfh.ch
Stefan Egli, eglis@bfh.ch

Berner Fachhochschule
Technik und Informatik

Version vom 18. März 2008



Betreuer: Urs Künzler, CPVR, BFH-TI,
urs.kuenzler@bfh.ch

Auftraggeber: Dr. Thilo Weitzel, Nuklearmedizin, Inselspital Bern,
thilo.weitzel@insel.ch

Experte: Dr. Federico Flückiger

* <https://www2.ti.bfh.ch/fbi/2008/Bachelor-Thesis/IKLU1-1-08-de.xml>

Inhaltsverzeichnis

1	Dokumentversionierung	1
2	Einleitung	2
3	Ausgangssituation	2
4	Bachelor Thesis - Aufgabenstellung	3
5	Zielbestimmungen	3
5.1	Musskriterien	3
5.2	Wunschkriterien	3
5.3	Abgrenzungskriterien	4
6	Applikationsanforderung (Produkt oder Prototyp)	4
7	Applikationseinsatz	5
8	Use Cases	5
8.1	Kurzbeschreibung der Use Cases	6
9	Qualitätsanforderungen	8
9.1	Funktionale Qualitätsanforderungen	8
9.2	Nichtfunktionale Qualitätsanforderungen	8
10	Application Data Management	8
11	Software Schnittstellen	9
12	Systemanforderungen für Entwicklung und Betrieb	9
13	Entwicklung	9
14	Verwendete Testverfahren	10
14.1	Testing auf der Core Ebene (Logik)	10
14.2	Testing auf der Applikationsebene (Frontend/GUI)	10
15	Projektmanagement	11
15.1	Organisation	11
15.2	Milestones	11
16	Glossar	12
17	Unterschriften	14

1 Dokumentversionierung

<i>Datum, Iteration</i>	<i>Phase</i>	<i>Wer</i>	<i>Was</i>	<i>Version</i>
29.02.2008, 1	Inception	Team Horus	Initial	0.1
06.03.2008, 1	Inception	vonnp	Use Cases, Testverfahren	0.1.1
07.03.2008, 2	Inception	vonnp	Ergänzung Pflichtenheftelemente	0.2
08.03.2008, 2	Inception	vonnp	Strukturänderungen, Ergänzungen	0.2.1
11.03.2008, 3	Inception	vonnp, eglis	Use Cases Refactoring	0.2.2
12.03.2008, 3	Inception	vonnp, eglis	Ergänzungen, Use Cases	0.2.3
13.03.2008, 4	Inception	Team Horus	Strukturänderungen, Korrekturen	0.2.4
14.03.2008, 4	Inception	vonnp	Strukturänderungen, Korrekturen	0.2.5
15.03.2008, 5	Inception	vonnp	Strukturänderungen, Korrekturen	0.2.6
18.03.2008, 5	Inception	Team Horus	Finale Version	1.0

2 Einleitung

Dieses Dokument stellt die Sektionen des Lasten- und Pflichtenhefts für unsere Bachelor Thesis 2008 dar. Der Einfachheit halber haben wir diese zwei Dokumente, die eigentlich getrennt sein sollten, gemischt, um eine einfache konsistente Sicht auf die Aufgabenstellung sowie die Ziele (die Lasten) und Wege zur fertigen Applikation (die Pflichten) zu ermöglichen. Des Weiteren sind auch Use Cases enthalten. Weitere Dokumente (Artefakte resp. Deliverables), die wir ebenfalls erstellen und jeweils ergänzen werden, sind eine Systemdokumentation, ein Projektmanagementdokument als Entwickler Wiki sowie ein Benutzerhandbuch.

Wir beziehen uns für die Gliederung dieses Dokuments auf die Seiten von Stefan Baur ^{1 2} und die deutsche Wikipedia ^{3 4}.

3 Ausgangssituation

Die kombinierte Positronen-Emissions-Tomographie (PET) / Computer-Tomographie (CT) ist eine nicht-invasive Bildgebungsmethode, die anatomische und metabolische Zusammenhänge in einem Untersuchungsgang erfasst. Seit der Einführung des PET/CT im Jahr 1998 hat sich diese Methode in der klinischen Anwendung rasant verbreitet und wurde ständig verbessert. Die Untersuchungsmethode liefert hochauflösende 3-dimensionale Datensätze, welche bisher mit konventionellen Methoden in Form von Schnittbildern dargestellt werden. Die ständig steigenden Datenmengen und Anforderungen nicht nur im Bereich der Diagnostik, sondern auch im Bereich der Planung chirurgischer Massnahmen oder der Bestrahlungsplanung, erfordern innovative Ansätze für die Interpretation und Bewertung der gewonnenen Daten.

¹ <http://www.stefan-baur.de/cs.se.lastenheft.html?glstyle=2004> - 29.02.2008

² <http://www.stefan-baur.de/cs.se.pflichtenheft.beispiel.html> - 07.03.2008

³ <http://de.wikipedia.org/wiki/Lastenheft> - 06.03.2008

⁴ <http://de.wikipedia.org/wiki/Pflichtenheft> - 07.03.2008

4 Bachelor Thesis - Aufgabenstellung

- Implementation von Routinen zum Einlesen der medizinischen Datensätze im DICOM-Format
- Implementation von Routinen zur Handhabung des stereoskopischen Displays
- Implementation von Routinen für die Darstellung eines stereoskopischen Volume Renderings
- Untersuchung der Möglichkeiten für den Einsatz eines Non-Photorealistic Renderings (NPR)
- Implementation einer Benutzeroberfläche

5 Zielbestimmungen

Mit Hilfe unserer Applikation *HORUS* soll es Fachpersonal ermöglicht werden, kombinierte PET/CT-Bildinformationen eines Patienten in 2D, 3D oder stereoskopisch zu betrachten und zu analysieren. Die Applikation soll intuitiv zu bedienen sein und sich optimal in den Arbeitsablauf des Fachpersonals integrieren.

5.1 Musskriterien

Die Musskriterien umfassen die folgenden Punkte der Bachelor Thesis:

- Implementation von Routinen zum Einlesen der medizinischen Datensätze im DICOM-Format
- Implementation von Routinen zur Handhabung des stereoskopischen Displays
- Implementation von Routinen für die Darstellung eines stereoskopischen Volume Renderings
- Untersuchung der Möglichkeiten für den Einsatz eines Non-Photorealistic Renderings (NPR)
- Implementation einer Benutzeroberfläche

5.2 Wunschkriterien

Die nachträglichen Wunschkriterien umfassen die folgenden Punkte:

- Speicherung von Benutzereinstellungen (Profileinstellungen wie Augenabstand für die stereoskopischen Viewer, Einfärbungen der Modelle, Algorithmen etc.)
- Multiplanare 2D View
- Einfrieren von Views (Sichten auf Modelle), ohne Speicherung von DICOM Daten
- Export von Views in Grafikformate (JPEG, PNG) für Ausdruck etc.
- CT Mesh Extraktion
- Navigation mit externen Devices resp. Head Tracking

5.3 Abgrenzungskriterien

Unsere Applikation gilt als *Standalone Application*, da sie keine weiteren direkten Abhängigkeiten zu Drittsystemen wie Datenbanken oder Mailservern hat. Sie ist einzig abhängig von der PET/CT Infrastruktur. Diese dient als Datenlieferant der DICOM Files und wird Software mässig nicht eingebunden. Ein Zugriff auf die produzierten Daten jedoch muss möglich sein (z. B. Dateizugriff via Netzwerk, lokale externe Harddisk, DVD-Datenträger etc.).

Das DICOM-Netzwerkprotokoll für den direkten Datenaustausch zwischen der PET/CT Infrastruktur und unserer Applikation wird nicht implementiert.

Folgende Illustration verdeutlicht die Grenzen (Boundaries) unseres Systems:

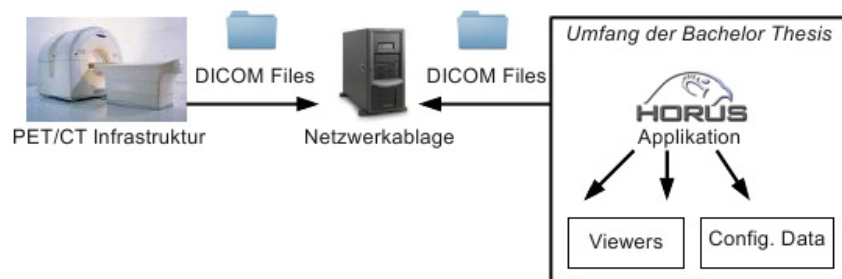


Abbildung 1. System Boundaries

6 Applikationsanforderung (Produkt oder Prototyp)

Die Aufgabenstellung der Bachelor-Arbeit erlaubt es uns, diesen Punkt selber zu definieren. Da es eher schwierig ist, aus dem Stand ein Produkt (bereit für Vermarktung und Verkauf inkl. Support etc.) zu lancieren, verstehen wir unsere Applikation als Prototyp im Umfang der Aufgabenstellung. In einer späteren Phase kann der Prototyp erweitert resp. redesigned werden, um die Grundlage für ein Produkt zu schaffen (Diplomarbeit).

Das Hauptziel besteht darin, effiziente Visualisierungsalgorithmen für die Darstellung von PET/CT Daten zu entwickeln. Diese sind in eine Applikation, die gute Benutzerführung bietet, gekapselt.

7 Applikationseinsatz

Die Anwendungsbereiche umfassen die 3D Unterstützung von prä-operativen radiologischen Untersuchungen im Inselspital Bern. Die Zielgruppe umfasst alle beteiligten Radiologen und Ärzte sowie MTA/MTRA.

Die Anwendung soll daher fehler- und wartungsfrei funktionieren und die gewünschten Resultate liefern; analog anderen Viewern in der medizinischen Bildbetrachtung. Da die Applikation nicht "mission critical" ist und kein Service Level Agreement besteht, sind keine weiteren Anforderungen an die Betriebsbedingungen gegeben.

Use Cases, die das Einsatzspektrum der Applikation aus der Benutzersicht definieren, sind in der nachfolgenden Sektion in diesem Dokument ersichtlich.

8 Use Cases

Die Use Cases decken die Musskriterien sowie die optionalen Wunschkriterien und das Einsatzgebiet der Applikation ab. Auf eine tabellarische Darstellung der Use Cases sowie auf alternative Workflows wird der Übersicht halber verzichtet.

Als *Actors (Primary)* gelten, wie eingangs erwähnt, die Radiologen, Ärzte und MTA/MTRA des Inselspitals Bern. Der Patient gilt in diesem Sinne nicht als *Actor*, da er lediglich die Datengrundlage des *System under Discussion (HORSUS)* bildet.

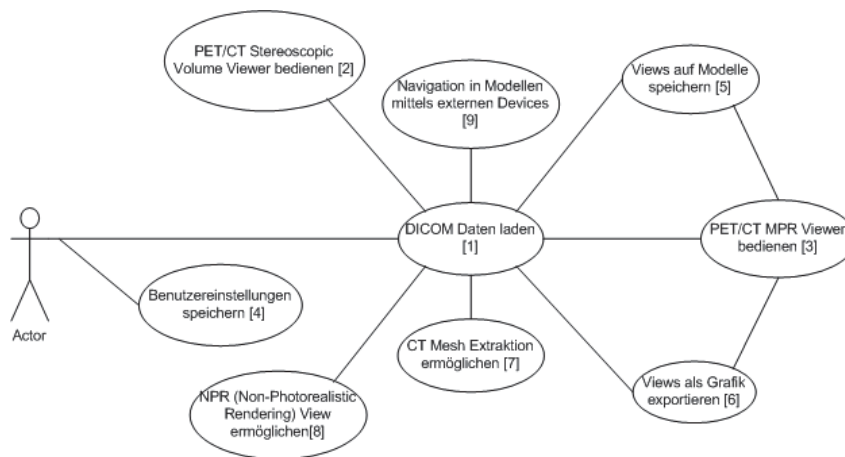


Abbildung 2. Use Cases

8.1 Kurzbeschreibung der Use Cases

[1] Benutzer lädt DICOM-Files inkl. Parametern

Es erfolgt die Auswahl des Datensatzes des entsprechenden Patienten durch den Benutzer über einen Dateiverwalter, welcher Zugriff auf den Datenbestand hat. Die ausgewählte Sitzung (PET/CT Durchgang) erscheint in der Selektion und kann in die Viewer exportiert werden. Zu jeder Zeit soll die Applikation terminierbar sein oder in den Grundzustand zurückversetzt werden können. Die Applikation stellt dabei sicher, dass nur Bilder des gleichen Patienten und der gleichen Studie in einen Datensatz gehören.

Vor dem eigentlichen Laden des Datensatzes in den Speicher können Angaben für ein *Preprocessing* gemacht werden:

- Ränder ohne Informationsgehalt können abgeschnitten werden
- Der Datensatz kann skaliert werden (Subsampling)
- Teilbereiche des Datensatzes (Slices) können selektiert werden, um nicht den ganzen Datensatz zu laden

[2] PET/CT Stereoscopic Volume Viewer bedienen

Nach erfolgreichem Laden der Datensätze kann der Benutzer mittels verschiedener Parameter den späteren Rendering-Prozess beeinflussen. Nach erfolgreicher Kalibrierung der benötigten Parameter wird der Rendering-Prozess vom Benutzer ausgelöst. Nach dem erfolgreichen Rendern der Szene kann der Benutzer das Modell im Raum mit der Tastatur bewegen.

In dieser View können folgende zusätzliche Funktionen ausgeführt werden:

- Kombinierte Darstellung von PET/CT
- Die Farbtransferfunktion kann angepasst werden
- Stereoskopische Sicht auf das Modell. Um einen vollumfänglichen 3D Effekt zu erzielen, benutzt der Anwender eine Shutter-Brille, welche von der Grafik-Hardware getriggert wird
- *Level of Detail* für das Volume Rendering anpassen (**optional**)
- *Clipping Plane* für das Volume Rendering beeinflussen (**optional**)

[3] PET/CT MPR (Multiplanare Rekonstruktion) Viewer bedienen

Nach erfolgreichem Laden eines Datensatzes können die Daten in einem MPR Viewer betrachtet werden (PET, CT oder kombiniert). Die einzelnen Bilder werden jeweils in der X-, Y- und Z-Achse (Seitenansicht, Frontansicht, Topansicht resp. sagittal und koronal) dargestellt und können schichtweise durchlaufen werden (z.B. mit Scroll-Wheel der Maus). Zu jeder Zeit soll die Applikation auf die Basisoberfläche zurückkehren können um andere Datensätze zu laden.

[4] Benutzereinstellungen speichern (optional)

Benutzer- und applikationsspezifische Parameter können via einen Wizard gespeichert und/oder geladen werden. Dieser Arbeitsschritt soll zu jeder Zeit, in der die Applikation nicht durch Berechnungen beschäftigt ist, möglich sein. Die Benutzereinstellungen beinhalten primär Parameter wie das Öffnen von Standardfenstern, Einfärbungen der Modelle, Laufwerke der DICOM Daten, individuelle Augenabstände für die stereoskopischen Viewer, Auswahl der Algorithmen etc.

[5] Views auf Modelle speichern (optional)

Der Benutzer kann aktuelle View für späteres Prozessieren/Analysieren speichern (Snapshots). Er kann für weitere Arbeitsschritte die aktuelle Sicht auf das Modell exportieren resp. speichern. DICOM Daten sollen dafür aber nicht gespeichert werden müssen. Bei erneuter Auswahl dieser Sicht für das entsprechende Modell wird diese angewendet, ohne dass jeder einzelne Arbeitsschritt bis zum gewünschten Aufriss wiederholt werden muss.

[6] Views als Grafik exportieren (optional)

Der Benutzer kann aktuelle View in ein Grafikformat exportieren (Screendump), damit diese für spätere Offline Arbeitsschritte zur Verfügung steht (Ausdruck, Nachbearbeitung in Bildbearbeitungsprogrammen etc.).

[7] CT Mesh Extraktion ermöglichen (optional)

Aus dem CT Bild soll ein Mesh extrahiert werden können. Das Modell soll in einem Viewer dargestellt werden und evtl. mit dem PET Volumen kombiniert werden können.

[8] NPR (Non-Photorealistic Rendering) View ermöglichen (optional)

Ein in den Speicher geladenes und dargestelltes Volumen kann nicht-photorealistisch dargestellt werden. Der Benutzer kann durch interaktives Verändern von Schwellwerten, Farb-Transferfunktionen und Materialeigenschaften die Art des Renderings beeinflussen. Mit diesen Techniken können Bereiche des Bildes farbig hervorgehoben werden. Durch ein möglichst grosses Spektrum von Einstellungen soll es möglich sein, eine künstlerisch-kreative oder illustratorische Darstellung zu erreichen.

[9] Navigation in Modellen mittels externen Devices (optional)

- Steuerung des Modells durch externe Devices. Der Benutzer kann die Applikation durch externe Geräte kontrollieren. Für eine intuitive Navigation kann der Benutzer z. B. eine Spacemouse verwenden
- Head Tracking. Hat der Benutzer ein 3D Modell generieren lassen, so kann er mit der entsprechenden Infrastruktur (Head Tracking Mechanismus) um das stereoskopische Modell wandern ohne weitere Input Devices für die Orientierung (Drehung) des Modells zu benötigen. Die Darstellung folgt der Position des Betrachters
- Einstellungen am Modell. Mittels USB MIDI Steuerungspulten können Parameter (z. B. Einfärbungen der Modelle) “on the fly” angepasst werden

9 Qualitätsanforderungen

9.1 Funktionale Qualitätsanforderungen

- Die Applikation verlangt viel Speicher. Die Robustheit der Algorithmen und die Stabilität des Betriebssystems muss daher auch unter diesen Einschränkungen gegeben sein (sicheres Memory-Management)
- Schnelle, einfache Profilkonfiguration (pro Benutzer)
- Die Darstellung muss exakt sein und visuell ansprechend
- Die Applikation soll modular so aufgebaut sein, dass gewisse Funktionen und Algorithmen auch durch Drittprojekte genutzt werden können

9.2 Nichtfunktionale Qualitätsanforderungen

- Die Applikation verletzt keine Arztgeheimnisse resp. die Patientendaten sind zu schützen
- Ursprungsdaten (DICOM) werden nicht manipuliert
- Schnelle, einfache Installation
- Die Benutzerführung soll intuitiv (selbsterklärend) sein ohne stete Konsultation des Benutzerhandbuches
- Die Applikation muss die Datensätze ”schnell” einlesen und darstellen können
- Die Portierbarkeit der Applikation beschränkt sich auf Linux und Windows (gleiche Hardware)

10 Application Data Management

Diese Sektion beschreibt die Daten, die von der Applikation gespeichert resp. für die Konfiguration eingelesen werden.

- Konfigurationsdaten resp. Benutzerdaten (Profileinstellungen wie Augenabstand für die stereoskopischen Viewer, Einfärbung der Modelle, Rendering Algorithmen etc.)
- Screendumps im JPEG- oder PNG-Format für die Ausgabe an einen Drucker
- DICOM Daten (resp. Patientendaten) werden nicht manipuliert und unterliegen den Sicherheitsbestimmungen des Inselspitals Bern

11 Software Schnittstellen

Unter Umständen kann mit unseren Partnerprojekten (PhD Arbeit an der Insel resp. **Registrierung der Patientenposition für PET/CT** ⁵) eine gewisse Codebase für engumgrenzte Aufgabengebiete geteilt werden. So z. B. das Einlesen von DICOM Daten mit vorhandenen eigenen Klassen.

12 Systemanforderungen für Entwicklung und Betrieb

- Betriebssystem (Windows XP SP2 resp. Linux 2.6.20 oder höher mit KDE 3.5 oder höher) mit OpenGL Unterstützung
- Grafikworkstation mit mindestens 2048 MB Hauptspeicher und 2 Ghz Intel Prozessor oder höher
- Nvidia Quadro FX 3400/4400 Grafikkarte mit mindestens 768 MB Onboard-Speicher
- SATA Harddisk (oder äquivalent)
- Netzwerkanschluss (100 MBit/s oder mehr) an den DICOM Datenstand

13 Entwicklung

Die Implementation unserer Applikation erfolgt gemäss der Aufgabenstellung mittels C/C++, OpenGL und Qt. Wir verwenden einen vereinfachten *Unified Process* mit starker Kundeneinbindung in die Entwicklung, um eine für die Zielgruppe zufriedenstellende Applikation herstellen zu können.

Frameworks

- OpenGL 2.1 resp. Mesa 7.0.2 (Grafikbibliotheken)
- GLSL für OpenGL 2.1 (Shader Language zu OpenGL)
- Qt 4.3.4 (GUI, Datenstrukturen & Algorithmen)
- DCMTK 3.5.4 (DICOM Handling Toolkit)
- Coin3D 2.5.0 (3D Graphics Development Toolkit)
- SOQt 1.4.1 (GUI Bindings für Coin3D)
- SIM Voleon 2.0.1 (Volume Rendering)
- OpenMesh 1.1.0 (Polygonal Meshes Support Toolkit)
- Boost C++ Libraries 1.34.1 (Datenstrukturen & Algorithmen)
- log4cplus 1.0.2 (Logging Facility)
- UnitTest++ 1.0 (Test Cases Framework)

⁵ <https://www2.ti.bfh.ch/fbi/2008/Bachelor-Thesis/ICTR1-4-08-de.xml>

Entwicklungsumgebung & Development Tools

- Eclipse 3.3.2 (Winter Edition)
- CDT 4.0.3 (C/C++ Plug-In für Eclipse)
- gcc 4.1.2 (GNU Compiler Collection)
- Cygwin (Linux ähnliche Umgebung für Windows)
- Eclox 0.6.5 (Eclipse Plug-In für Doxygen)
- Subclipse 1.2.4 (Eclipse Plug-In für Subversion)
- Valgrind 3.3.0 & Valkyrie 1.2.0 (Memory Leak Checkers)
- Subversion 1.4.6 (SCM Server)
- Pulse (Continuous Integration and Reporting Server)
- GDB 6.6 & DDD 3.3.11 (Debuggers)
- Gobby 0.4.5 (für kollaborative Code Sessions)

Dokumentationstools

- MediaWiki auf externem Server als Projekthomepage und für das Projektmanagement (siehe dazu Sektion **15**)
- Doxygen 1.5.2 (Code Dokumentation)
- L^AT_EX 1.40.3 (Web2C 7.5.6) (offizielle Dokumente)
- Kile 2.0.0 (LaTeX Frontend)
- Umbrello 2.0.0 (UML Diagramm Editor)
- Microsoft Visio 2007 (UML Diagramm Editor)
- OpenOffice Suite 2.3.0 (anderweitige Dokumente)

Weiteres

- Skype (VoIP) (Teamkollaboration)
- GMail (Hauptkommunikationsplattform)
- Interner Server der BFH als Backup für SCM, MediaWiki und Tests

14 Verwendete Testverfahren

14.1 Testing auf der Core Ebene (Logik)

Die Entwickler verwenden Unit Tests auf Modulebene mit den Bibliotheken Unit-Test++ und log4cplus. Desweiteren kommt ein Code Integration Server zum Einsatz, welcher nach jedem Check-In in das SCM (Subversion) automatisch Testskripts anstösst. Jedes Modul wird auf die Hauptfunktionen hin getestet. Die Use Cases dienen dabei als Grundlage. Der Einsatz von Exceptions soll ein unkontrolliertes Verhalten der Applikation im Fehlerfalle verhindern.

14.2 Testing auf der Applikationsebene (Frontend/GUI)

Die in den Use Cases beschriebenen Vorgänge und zu erreichenden Goals müssen gemacht werden können resp. ersichtlich sein. Die Tests erfolgen durch die Entwickler sowie den Kunden und die beteiligten Fachkräfte. Einfache GUI Tests können auch mit dem von Qt mitgelieferten Testing Framework automatisiert werden.

15 Projektmanagement

15.1 Organisation

Da das Projektmanagement ein dynamischer Prozess ist, haben wir uns entschlossen, die Dokumentführung auf einem Wiki abzuwickeln.

Dies bringt uns folgende Vorteile:

- Jeder Entwickler hat zu jeder Zeit die volle Information über das Projekt
- Vereinfachte Organisation und zentraler Meeting-Point
- Übersicht über Milestones, aktuelle Tasks, Iterationen und Termine
- Gleichzeitiges Bearbeiten (Kollaboration) der Inhalte

Desweiteren ist ein Code Integration Server vorhanden, der uns in obigen Punkten ebenfalls zur Verfügung steht.

Bitte *hier*⁶ klicken, um die Projekthomepage zu öffnen.

15.2 Milestones

Folgende Milestone-Daten wurden bereits festgelegt:

- 03.04.2008, zu erreichen: DICOM Reader, Logik (inkl. Exceptions), GUI und Profilverwaltung (ansatzweise)
- 10.04.2008, zu erreichen: Evaluation Rendering Verfahren
- 17.04.2008, zu erreichen: Volume Rendering (ansatzweise)
- 15.05.2008, zu erreichen: Stereoscopic Volume Viewer (ansatzweise)
- 13.06.2008, 17:00 Uhr Abgabetermin der Bachelorarbeit

Weitere Milestones sind Gegenstand der iterativen Planung unseres Projekts.

⁶ <http://node331.ww2.ch/mediawiki>

16 Glossar

- *Clipping Plane*, Bereich des Modells, der durch den Rendering Prozess nicht berücksichtigt wird
- *Continuous Integration*, Prozess oder Infrastruktur der permanenten Code Überwachung (Testing)
- *CT*, Computertomographie, vertikales Schichtröntgen mit Computerunterstützung
- *DICOM*, Digital Imaging and Communications in Medicine, Datenformat in der Medizin
- *GUI*, Graphical User Interface, Graphische Benutzeroberfläche
- *Head Tracking*, Rückverfolgung der Kopfposition resp. von Kopfbewegungen
- *HORUS*, Horus, Altägyptische Gottheit des Himmels und des Lichts, Sohn des Osiris
- *Level Of Detail*, Qualitative Angabe der Berücksichtigung von Details des Renderings
- *Mesh*, Primitive Struktur in der Computergraphik, bestehend aus Knoten, Kanten und Flächen
- *MIDI*, Musical Instrument Digital Interface, Schnittstelle zur Steuerung von Softwaremodulen mittels Keyboards etc.
- *MPR*, Multiplanare Rekonstruktion von Schichtdaten des menschlichen Körpers
- *MTA/MTRA*, Medizinisch-Technischer Assistent/Medizinisch-Technischer Röntgenassistent
- *Multiplanar*, Berechnung von sagittalen resp. koronalen Bilddaten aus Schichtdaten
- *PET*, Positronenemissionstomographie, Nuklearmedizinisches Bildgebungsverfahren zur Visualisierung von Stoffwechselfvorgängen
- *Service Level Agreement (SLA)*, Abkommen über zu erbringende Leistungen im Problemfall
- *Stereoskopisch*, Reale 3D Darstellung, erfordert technische Zusatzgeräte wie Shutter-Brillen etc. im Gegensatz zu Pseuo 3D auf 2D Bildschirmen

- *System Boundaries*, Implementierungsgrenzen eines (Software)-Systems
- *Unified Process*, Iterativer, inkrementeller Software Entwicklungs Prozess, im Gegensatz zum Wasserfallmodell
- *Use Case*, Verhalten eines Systems unter benutzerabhängigen Vorgehensweisen
- *View*, Sicht, Anzeigepanel für Graphiken
- *Volume Rendering*, Berechnung und graphische Darstellung eines 3D Modells aus 2D Daten

17 Unterschriften

Abnahme des Dokuments.

Der *Betreuer*:

Der *Auftraggeber*:

Die *Studenten*:

Ort, Datum: