



Zellomat3D

Projekt Management Plan

Projekt: 3D Cellular Automata Simulator – Diplomarbeit – SS/2005

Auftraggeber: Hochschule Rapperswil HSR

Betreuer: Eduard Glatz – Prof. Dipl. Ing. ETH eglatz@hsr.ch

Mitarbeiter: Michael Florin loop@loop.li
Andreas Weinmann a.weinmann@gmx.ch

Ablage: ProjektManagementPlan - 16032005.doc



Inhaltsverzeichnis

EINFÜHRUNG	4
ZWECK	4
GÜLTIGKEITSBEREICH	4
INHALT	4
PROJEKT BESCHREIBUNG.....	5
PROJEKTÜBERSICHT	5
LIEFERUMFANG.....	5
ENTWICKLUNG DES PROJEKTPLANS	5
QUALITÄTSSICHERUNG	6
STYLEGUIDE	6
PROJEKTORGANISATION.....	7
PROZESSMODELL.....	7
ORGANISATIONS-STRUKTUR.....	7
KONTAKTANGABEN.....	8
ABGRENZUNG UND SCHNITTSTELLEN	8
PROJEKTMANAGEMENT.....	9
MANAGEMENT ZIELE UND PRIORITÄTEN	9
FUNKTIONSUMFANG	9
TIME MANAGEMENT	9
PERSONALPLANUNG.....	9
ÜBERWACHUNG UND KONTROLLE	9
REVIEWS.....	10
VERSIONSKONTROLLE.....	10
RISK MANAGEMENT	10
RISK MANAGEMENT	10
ALLGEMEINE RISIKEN.....	11
TECHNISCHE RISIKEN.....	12
TECHNISCHE RISIKEN.....	13
TECHNISCHE RISIKEN.....	14
ENTWICKLUNGSPROZESS	15
TECHNOLOGIEN	15
TOOLS.....	15
SDK	15
OS.....	15
DOKUMENTATION.....	16
PROJEKT -NTERSTÜTZUNG	16
ENTWICKLUNGSPLAN.....	17
ARBEITSPAKETE.....	17
1. PHASE VORBEREITUNG	17
2. PHASE ANALYSE	18
3. PHASE DESIGN	18
4. PHASE IMPLEMENTATION.....	18
5. PHASE ABSCHLUSS.....	18
ARBEITSUMGEBUNG.....	19
AP1.1	19
AP1.2	19
PROJEKTMANAGEMENT	20
AP1.3	20
UAP1.4	20
AP1.5.....	21



UAP1.6	21
AP1.7	22
UAP1.8	22
UAP1.9	22
ANALYSE	23
AP2.1	23
AP2.2	23
DESIGN	24
AP3.1	24
AP3.2	25
IMPLEMENTATION 1. ITERATION.....	26
AP4.1	26
AP4.2	26
AP4.3	26
IMPLEMENTATION 2. ITERATION.....	27
AP4.4	27
AP4.5	27
AP4.6	27
AP4.7	28
AP4.8	28
BERICHTE.....	29
AP5.1	29
AP5.2	29
AP5.3	30
AP5.4	30
REFERENZEN.....	31
UAP5.5	31
UAP5.6	31
UAP5.7	31
HANDBÜCHER.....	32
AP5.8	32
AP5.9	32
ABSCHLUSS.....	33
AP5.10	33
AP5.11	33
TERMINPLAN	34
MEILENSTEINE	34
ZEIT AUSWERTUNG.....	35
PROJEKTSTUNDEN PRO WOCHE	35
PROJEKTSTUNDEN AUFSUMMIERT PRO WOCHE	36
PROJEKTSTUNDEN NACH PHASEN.....	37
TEAMMITGLIEDER STUNDEN PRO WOCHE.....	38
TEAMMITGLIEDER STUNDEN AUFSUMMIERT	39
ZEIT PRO TÄTIGKEITSBEREICH	40
ZEIT PRO PHASE	41
CODESTATISTIK	42
GESAMTÜBERSICHT	42
GRAPHIK ZEILENSTATISTIK	42



Einführung

Dieses Dokument regelt organisatorische Belange. Die **Zweck** Organisationsstruktur, die Ziele und die Entwicklungsmethoden sind festgehalten. Es dient als Richtlinie und als Arbeits- bzw. Richtungsvorgabe für die Mitarbeiter. Ausserdem eröffnet es dem Betreuer einen Einblick in die Arbeitsweise des Entwicklerteams.

Dieses Dokument gilt für die Diplomarbeit "Zellomat3D", welche **Gültigkeitsbereich** im SS/2005 an der Hochschule Rapperswil HSR durchgeführt wurde.

Im Projekt Management Plan werden sowohl die **Inhalt** Randbedingungen des Projekts als auch das Projektmanagement, die Projektorganisation und der Entwicklungsprozess beschrieben. Zudem wird ein Überblick über die ganze Diplomarbeit vermittelt.



Projektbeschreibung

Die Theorie der Cellular Automata befasst sich seit längerem mit selbst wachsenden Organismen auf einer computertechnischen bzw. mathematischen Basis. Praktische Anwendungen sind bei adaptiven und selbstoptimierenden Systemen zu finden, die im Autonomic Computing eine grosse Rolle spielen. Jedoch nur schon die Visualisierung selbstwachsender Systeme ist von Interesse.

Projektübersicht

Der Zellomat3D beinhaltet verschiedenste Funktionen, um die Berechnung und die visuelle Darstellung von verschiedenen zellulären Automaten zu unterstützen. Er ermöglicht es, auf einfache Art und Weise selbst erstellte Automaten auf ihre Funktionsweise und Auswirkungen hin zu untersuchen. Durch eine hohe Verarbeitungsgeschwindigkeit und eine ansprechende 3D Visualisierung der einzelnen Entwicklungszyklen des Automaten können schnell und einfach die Wechselwirkungen der zugrunde liegende Regeln beobachtet werden.

Am Schluss des Projekts erwartet der Auftraggeber eine funktionsfähige Applikation inklusive Quellcode sowie die dazugehörige Dokumentation. Dokumentationen sind elektronisch sowie auf Papier abzuliefern. Der Source Code ist nur in elektronischer Form beizulegen. Eine Übersicht der abzugebenden Dokumente ist weiter unten „Dokumente“ zu finden.

Lieferumfang

Der Experte erhält die Arbeit im gleichen Umfang wie der Auftraggeber, um sie zu bewerten. Der genaue Lieferumfang der Software wird in der Anforderungsspezifikation¹ beschrieben. Die abzugebenden Dokumente, Software und zusätzliche Komponenten der Diplomarbeit, werden auf eine Projekt-CD gebrannt und auf einer Projekt-Homepage verlinkt und dort mit einigen zusätzlichen Informationen übersichtlich dargestellt.

Der Projektplan wird kontinuierlich angepasst und als eigenständiges Dokument geführt. Die Ersteinführung dieses Projektplans wird auf den 16. März 2005 festgelegt.

Entwicklung des Projektplans

¹ Zu finden in der Datei Anforderungsspezifikation – 21032005.doc



Jede Woche findet eine Sitzung mit dem Auftragsgeber statt. Die offiziellen Reviews werden bei den Sitzungen durchgeführt. Diese finden vornehmlich montags um 14:00 in der HSR Rapperswil, im Büro von Herrn Glatz statt. Von den wöchentlichen Sitzungen wird ein Stichwortprotokoll erstellt sowie nachträglich ein ausführliches Sitzungsprotokoll. **Qualitätssicherung**

Bei ausserordentlichen Zwischenfällen werden wir auch Ad-hoc Meetings veranstalten oder offene Fragen per Email oder Telefon klären. Kleinere Reviews in der Projektgruppe finden jeweils bei Bedarf in der HSR oder zu Hause statt und werden nur protokolliert, wenn grössere Mängel oder schwerwiegende Probleme festgestellt wurden.

Nachdem ein Dokument von einem Projektmitglied verfasst wurde, wird es vom Partner in einem kleinen Review² geprüft und korrigiert. Somit sind alle Parteien gleich stark für die Korrektheit, Vollständigkeit und Qualität der Dokumentation verantwortlich. Der implementierte Code wird ebenfalls gegenseitig kontrolliert und verbessert. Auch für den Code findet jede Woche ein kurzes Review statt. Es wird ausserdem ein Bugtracking zur Qualitätssteigerung eingesetzt.

Alle Entwickler halten beim Programmieren an den Styleguide **Styleguide** (beschrieben in StyleGuide - 16032005.doc) ein.

² Teaminterne Reviews werden keine Protokolle geführt. Die Änderungen werden direkt vorgenommen. Falls dabei wichtige Änderungen vorgenommen werden, werden diese in der Versionskontrolle der Analyse- oder Designdokumente vermerkt.



Projektorganisation

Die Software wird nach dem Rational Unified Process³ **Prozessmodell** entwickelt. Einzelne Dokumente⁴ werden dabei weggelassen. Zu erwähnen ist, dass wir die Software-Entwicklung in zwei Iterationen durchführen, um mit evolutionären Prototypen arbeiten zu können. Für mehr als zwei Iterationen ist der zeitliche Rahmen zu knapp.

Das Projekt wird von Michael Florin und Andreas Weinmann **Organisations-** bearbeitet und durch Eduard Glatz⁵ betreut. Er ist auch der **Struktur** Auftraggeber des Projektes. Herr Roberto Pajetta ist der Experte.

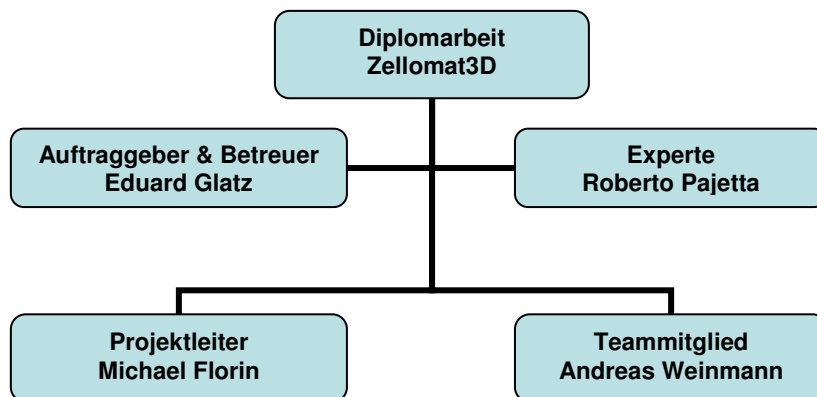


Abbildung 1: Organisationsstruktur

³ Rational Unified Process (RUP): speziell für das Softwareengineering entwickeltes Prozessmodell

⁴ Eine Liste der erstellten Dokumente ist in der Datei „DokumentenPlan – 05052005“.doc zu finden

⁵ Eduard Glatz, Professor für Computersysteme an der Hochschule Rapperswil HSR



Auftraggeber:
& Betreuer: Hochschule Rapperswil HSR
Eduard Glatz
Dipl. Ing. ETH
Oberseestrasse 10
8640 Rapperswil
Tel: 055 222 49 04
Mobile: 079 224 81 86
Email: eglatz@hsr.ch

Kontaktangaben

Experte: Roberto Pajetta
Dipl. Ing. ETH
Im Tiergarten 44
8055 Zürich
Tel: 01 462 40 30

Projektteam: Michael Florin
Dorfstrasse 7
8715 Bollingen
Tel: 079 66 33 001
Email: loop@loop.li

Andreas Weinmann
Dorfstrasse 7
8715 Bollingen
Tel: 079 542 47 41
Email: a.weinmann@gmx.ch

Dieses Projekt wird im Rahmen einer Diplomarbeit an der Hochschule Rapperswil durchgeführt. **Abgrenzung und Schnittstellen**
Die Qualitäts-Sicherung wird durch regelmässige Reviews und Tests garantiert.
Die Dokumentation wird hauptsächlich in deutscher Sprache verfasst. Das Userinterface, sowie der SourceCode werden in englischer Sprache verfasst. Die Code Kommentare in Deutsch.



Projektmanagement

Das Ziel des Projektmanagements ist es, den Projektverlauf so zu planen und zu steuern, dass die Vorgaben eingehalten werden können. Der Zeitrahmen ist fix. Ein kleines finanzielles Budget ist bei grossem Bedarf denkbar, falls Bücher oder kleinere Tools angeschafft werden müssen. Arbeitsstunden werden nicht verrechnet, aber dokumentiert. Ausserordentliche Aufwendungen müssten nur durch risikobedingte Vorkommnisse getätigt werden. Der einzige variable Parameter in der Projektierung ist der Funktionsumfang. Kurz: Die Termine und das Budget sind während des Projektes nicht veränderbar, einzig unsere Leistung muss auf die Einhaltung dieser Punkte hin ausgelegt werden.

Management Ziele und Prioritäten

Da der Funktionsumfang der einzige Faktor ist, auf den das Projektteam direkt Einfluss nehmen kann, legen wir darauf besonderen Wert. Er sollte aus verständlichen Gründen nicht zu klein ausfallen und er jedoch viel zu gross angesetzt, scheitert das Projekt.

Funktionsumfang

Die Zeit wird mittels Excel erfasst, damit der Soll und Ist-Fortschritt des Projekts jederzeit ersichtlich ist.

Time Management

Die Studienarbeit wird im Zweier-Team durchgeführt. Die Arbeiten werden in Arbeitspakete aufgeteilt und danach von den Mitarbeitern bearbeitet.

Personalplanung

Der geplante wöchentliche Aufwand beträgt zirka 66 Stunden pro Mitarbeiter. Bei Eintritt eines Risikos werden die eingeplanten Reservestunden benützt oder zusätzliche Stunden eingesetzt.

Die Teammitglieder arbeiten zu Hause oder in der HSR. Damit die Kommunikation optimal gewährleistet ist, werden zusätzlich ein Voice Over IP Tool (Skype⁶), E-mails sowie eine Online Kommunikationsplattform (Homepage⁷) eingesetzt.

Zur Überwachung werden die selbst erstellten Terminpläne und Zieldefinitionen, sowie die Sitzungsprotokolle benützt. Auf dem Version Controlling System (CVS)⁸ befinden sich alle Dokumente, sowie alle Versionen der bereits implementierten Softwaremodule.

Überwachung und Kontrolle

⁶ <http://www.skype.com>

⁷ <http://www.loop.li>

⁸ In diesem Projekt wurde zur Versionskontrolle WinVCS und CvsNT-Server eingesetzt



Regelmässige Sitzungen mit Herrn Glatz bieten eine zusätzliche **Reviews** Standortbestimmung. Jede Projektsitzung muss protokolliert werden. Somit kann der Fortschritt kontrolliert und allfällige Probleme können frühzeitig entdeckt werden. Das kleine Review wird indirekt mittels Änderungen an den entsprechenden Dokumenten festgehalten.

Die Software wird mit Unterstützung eines Concurrent Versions System (CVS) entwickelt. Die Dokumentation wird dort ebenfalls abgelegt, damit die ganze Arbeit jederzeit als Bundle zur Verfügung steht. **Versionskontrolle**

Dokumente (*.doc, *.xls, *.mpp, usw.), werden nach folgendem Schema benannt:
<DokumentName> - <Datum[DDMMYYYY]>.<Dateierweiterung>
Beispiel: Dokumentation -14032005.doc

Risk Management

In den folgenden Tabellen werden eine Übersicht der **Risk Management** verschiedenen Risiken, deren Auswirkungen sowie möglicher Lösungen aufgelistet.

Legende zur Wahrscheinlichkeit und Auswirkung:

- 1: gering
- 2: mittel
- 3: hoch



Allgemeine Risiken

Tabelle 1: Risikoabschätzung Allgemeine Risiken

Risikoquelle	Auswirkungen		Risiko		Vorkehrung/Massnahme
	Wahrscheinlichkeit	Auswirkung	Auswirkung	Risiko (Produkt)	
Keine oder verspätete Bestätigung / Beantwortung von Mails	1	1	1	1	Mailbestätigung anfordern und Annahmen treffen
Zeitverzögerung durch Auftraggeber	1	1	1	1	Projektumfang verkleinern. Antworten ausdrücklich verlangen
Kommunikationsprobleme innerhalb des Teams	1	2	1	2	Probleme mit dem Betreuer besprechen
Ausfall / Absenz Projektmitglied	2	3	2	6	Absenzen soweit möglich einplanen; Reserven in Terminplan vorsehen
Netzwerkausfall	1	2	1	2	Lokale Kopien benutzen und zu einem späteren Zeitpunkt Synchronisation nachholen
Rechnerausfall Schule	1	2	1	2	Ausweichmöglichkeiten vorsehen (Notebook, zu Hause)
Datenverlust	1	2	1	2	Backup, Datensicherung, Recovery
Änderung der Aufgabenstellung	1	3	1	3	Sorgfältige Analyse und Abgrenzung des Projektumfangs
Schwerwiegende Fehler in Analyse oder Design	1	3	1	3	Sorgfältige Analyse und Design
Technologische Schwierigkeiten	2	2	2	4	Reserven in Terminplan vorsehen, frühzeitig mit den schwierigen Teilen beginnen



Technische Risiken

Tabelle 2: Risikoabschätzung Technische Risiken

Risikoquelle	Auswirkungen			Risiko			Vorkehrung/Massnahme
	Wahrscheinlichkeit	Auswirkung	Risiko (Produkt)	Wahrscheinlichkeit	Auswirkung	Risiko (Produkt)	
Unerwartete Fehler durch unreife Technologien (z.B. DirectX 9.0 ist noch nicht vollkommen ausgereift)	1	2	2	1	2	2	APIs und Informationen dazu genau studieren, um eventuelle Fehler frühzeitig zu erkennen, nicht erst wenn das Entwicklungsstadium des Prototypen weit fortgeschritten ist
Unvollständige oder schlechte Dokumentationen der API's	2	1	2	2	1	2	Möglichst viele Informationen zu den einzelnen API's eventuell im Internet als Zusatzliteratur suchen. Zeit für Ausprobieren einplanen
Hardware-spezifische Informationen nicht zugänglich (Firmengeheimnisse, Grafikkchips, GPU's, Videobuffer usw.)	1	1	1	1	1	1	Durch frühzeitiges ausprobieren, ob man die Funktionen realisieren kann oder nicht, bevor das Entwicklungsstadium des Prototypen zu weit fortgeschritten ist und man erst dann in einer Sackgasse landet
Grosse Unterschiede bei der Grafikkhardware (API Unterstützungen)	3	2	6	3	2	6	Bei der Initialisierung prüfen, ob gewünschte Modi der Grafikkarte verfügbar sind und entsprechend im Programmablauf darauf reagieren
„Verrennen“ in Detailspekte	1	2	2	1	2	2	Kurze Zeit ausprobieren ob die gewünschte Funktionalität realisiert werden kann. Abbruch des Versuches nach 1-2 Tagen
Sehr schweres Debuggen, komplexe rekursive Algorithmen, (tausende Zyklen im Programmablauf)	3	2	6	3	2	6	Sauberer Planen des Designs und der Algorithmen, Reserven einplanen, Debugger sinnvoll einsetzen können (erlernen der Möglichkeiten)
Auftreten von Memoryleaks zu unbestimmten Zeitintervallen, gleiches Problem mit Debuggen bei zigtausenden Zyklen	3	2	6	3	2	6	Sauberer Planen des Designs und der Algorithmen, Reserven einplanen



Technische Risiken

Tabelle 3: Risikoabschätzung Technische Risiken

Risikoquelle	Auswirkungen			Vorkehrung/Massnahme		
	Wahrscheinliche	Auswirkung	Risiko (Produkt)	Wahrscheinliche	Auswirkung	Risiko (Produkt)
Rohdatengenerierung, ohne Möglichkeit die Daten visuell zu überprüfen (Consoleauswertung extrem mühsam bei tausenden Zyklen)	2	2	4	möglichst gleichzeitiges Entwickeln der Visualisierung und Rohdatengenerierung, eventuelle visuelle Ausgabe ohne Performanceaspekte (Hauptsache man sieht etwas)		
Durch "überoptimieren" des Codes können Performanceverluste entstehen.	2	1	2	Genaues durchdenken der Abhängigkeiten innerhalb des Programmablaufes unter dem Aspekt, das gewisse Optimierungen einen Nachteil für den nachfolgenden Programmablauf bringen könnten.		
Durch Vereinfachung des sequenziellen Codedurchlaufes kann es wiederum zu Performanceverlusten führen (Inlining, Delegates) von Funktionen	1	1	1	Genügend Kenntnisse über Codeoptimierungen aneignen (Code Profiling verwenden, viel ausprobieren und vergleichen der Laufzeiten)		
Gefahr der Objektorientierten Applikations-Entwicklung.	2	1	2	Genaue Untersuchung nötig, um den Performanceverlust durch den Objektorientierten Ansatz möglichst klein zu halten		
Gefahr von unverständlichem „Spagatticode“ durch Performanceoptimierungen.	3	1	3	Gesundes Mittel finden, sowie die Abläufe in der Dokumentation genau beschreiben.		
Gefahr der Inkompatibilität der einzelnen Module untereinander	1	3	3	Schnittstellendefinition im vornherein sauber definieren		
Datentransfer zwischen den einzelnen Modulen / Hardware	2	2	4	Datenmengen durch Optimierungen möglichst klein halten		
Doppeltspürigkeiten die zu Performanceverlusten führen können (z.B. ein grosses Array in zwei verschiedenen Modulen durchiterieren)	1	2	2	Durch das oberstehende Risiko bzw. dessen Lösung kann es ggf. zu Einbussen der Leistung kommen. Daher muss der Gesamtaspekt der Optimierung immer im Hinterkopf gehalten werden.		



Technische Risiken

Tabelle 4: Risikoabschätzung Technische Risiken

Risikoquelle	Auswirkungen			Risiko			Vorgehensmassnahme
	Wahrscheinlichkeit	Auswirkung	Risiko (Produkt)	Wahrscheinlichkeit	Auswirkung	Risiko (Produkt)	
Nachwirkungen von „über“ Optimierungen in den einzelnen Modulen, die sich auf die Performance des nachfolgenden Moduls auswirken kann. (Exponentiell)	Fehler durch „überoptimierung“ zieht sich durch den ganzen Programmablauf hindurch, was zu schweren Leistungseinbussen führt	1	3	3	Dieses Problem fällt auch in die Kategorie der "Gesamtüberblick bewahren" wenn man Optimierungen durchführt. Zu beachten sind die Auswirkungen die eine Optimierung auf nachfolgende Programmabläufe hat		
Performanceverlust durch zu feste Fokussierung der Optimierung der einzelnen Module, anstelle Optimierung über den gesamten Programmablauf.	Performanceverluste innerhalb des Programmablaufes	1	2	2	gleiche Lösung wie im vorhergehenden Problem		
Prototypen, die sich im nachhinein als unbrauchbar erweisen	Zeitverluste durch Aufwand in der Programmierarbeit	1	3	3	Überlegungen in der Analyse genau unter die Lupe nehmen, was ist möglich, was wollen wir und was wird überhaupt gefordert		
Hardwarebeschränkungen über alle Module hinweg beachten (z.B. Grosser Kubus kann generiert werden, aber nicht visualisiert werden)	Fehlerquellen innerhalb der Applikation	1	3	3	Auswirkungen auf die Daten, deren Generierung, Aufbereitung und Visualisierung über den gesamten Programmablauf berücksichtigen. Sauberes Exception-Handling		
Optimierung für die meisten Anwendungsfälle gut, aber im „Worstcase“ zu viel Overhead generiert (z.B. suboptimaler Automat)	Performanceverluste oder Absturz der Applikation	3	3	9	Kombination der verschiedenen Algorithmen und Programmabläufe berücksichtigen unter dem Aspekt des "Worst Cases"		



Entwicklungsprozess

C#	C# ist eine Programmiersprache, die vom Softwarehersteller Microsoft im Rahmen seiner .NET-Initiative eingeführt wurde.	Technologien
Managed DirectX 9.0	DirectX ist eine Sammlung von APIs (Application Programming Interfaces) für Multimedia-Programme (vor allem Computerspiele) auf den Windows-Betriebssystemen der Firma Microsoft.	
HTML&CSS	Wird für den Einsatz in der zusätzlichen Projekt Homepage verwendet.	
MS VS .Net 2003	Microsoft's Visual Studio .NET 2003 ist eine Entwicklungsumgebung für den Einsatz von C#	Tools
MS Office 2003	Word, Excel, Outlook, Project & Visio kommen in diesem Projekt zur Anwendung	
CVSNT WinCVS	Content Versioning Server (Version 2.0.58d) Content Versioning Client (Version 1.3.20.2)	
Rational Rose 2003	Software Engineering, dient zum Erstellen der Diagramme	
XMLSpy 2004	XML Analyse und Erstellung	
ANTSProfiler	Code Profiler (Version 2.0)	
Skype	Voice Over IP Kommunikationstool (Version 1.1)	
MeshView	Programm zur Anzeige von Meshes, im SDK von Managed DirectX 9.0 mit enthalten.	
.NET Framework	Software Development Kit für die Entwicklung von .Net basierten Applikationen (Version 1.1)	SDK
DirectX 9.0	Software Development Kit für das Erstellen von Applikationen, die 3D Grafikausgaben beinhalten.	
MS Windows 2000 & XP das	Das Betriebssystem aus dem Hause Microsoft dient für unser Projekt als Plattform für die Weiterentwicklung sowie für die Überprüfung und Deployment der entwickelten Software.	OS



Für die Dokumentation wurden Vorlagen erstellt, um ein **Dokumentation** einheitliches Erscheinen zu garantieren.

Die Dokumentation soll folgende Teile enthalten:

- Aufgabenstellung

- Projekt Management Plan
- Projektplan
- Zeiterfassung
- Style Guide
- Protokolle

- Anforderungsspezifikation
- Analyse
- Studie Algorithmen
- Design Prototypen
- Design Applikation
- Testbericht

- Management Summary
- Technischer Bericht
- Kurzfassung / Abstract
- Persönliche Berichte

- Glossar
- Literaturverzeichnis

- Benutzerhandbuch

Die einzelnen Teile werden in einem Gesamtbericht mit fortlaufender Seitennummerierung zusammengefasst.

Die Versionskontrolle wird mit Hilfe von CVSNT & WinCVS **Projekt -** realisiert und durch Michael Florin kontrolliert. Die SW-Qualitäts-**unterstützung** Sicherung erfolgt durch Reviews des Projektteams. Verifikation und Validation erfolgen am Ende jeder Phase.



Entwicklungsplan

Es sind zwei verschiedene Arten von Arbeitspaketen definiert. **Arbeitspakete** Die normalen Arbeitspakete (AP) beginnen während des Projektes und können nach einer gewissen Anzahl von Arbeitsstunden abgeschlossen werden. Die unterstützenden Arbeitspakete (UAP) werden während des gesamten Projektverlaufs bearbeitet.

Die geschätzten sowie die effektiven Arbeitsstunden der einzelnen Arbeitspakete sind aus dem Dokument „Zeiterfassung – 05052005.xls“ ersichtlich.

Die Diplomarbeit ist in fünf Phasen aufgeteilt, wobei die zweite bis vierte Phase in zwei Iterationen durchlaufen wird. Die erste Iteration dient der Erstellung evolutionärer Prototypen, die zweite der Implementation der Applikation.

Übersicht der einzelnen Phasen:

- **1. Phase:** Vorbereitung
- **2. Phase:** Analyse (zwei Iterationen)
- **3. Phase:** Design (zwei Iterationen)
- **4. Phase:** Implementation (zwei Iterationen)
- **5. Phase:** Abschluss

#	Beschreibung	Leiter
Arbeitsumgebung		
AP1.1	Software Installation	ALL
AP1.2	Templates	FLO
Projektmanagement		
AP1.3	Aufgabenstellung	WIM
UAP1.4	Projekt Homepage	FLO
AP1.5	Projekt Management Plan	FLO
UAP1.6	Projektplan / Zeiterfassung	ALL
AP1.7	Style Guide	WIM
UAP1.8	Protokolle	WIM
UAP1.9	Sitzungen	ALL

1. Phase Vorbereitung



#	Beschreibung	Leiter
Analyse		
AP2.1	Anforderungsspezifikation	WIM
AP2.2	Analyse (Prototypen&Applikation)	ALL

2. Phase Analyse

#	Beschreibung	Leiter
Design		
AP3.1	Design (Prototypen&Applikation)	ALL
AP3.2	Testspezifikationen	ALL

3. Phase Design

#	Beschreibung	Leiter
Implementation 1. Iteration		
AP4.1	Prototypen Modul Rohdatenberechnung	WIM
AP4.2	Prototypen Modul Datenaufbereitung	FLO
AP4.3	Prototypen Modul Visualisierung	FLO
Implementation 2. Iteration		
AP4.4	Modul Rohdatenberechnung	WIM
AP4.5	Modul Datenaufbereitung	WIM
AP4.6	Modul Visualisierung	FLO
AP4.7	Deployment Programm	ALL
AP4.8	Automatenentwicklung	WIM

4. Phase Implementation

#	Beschreibung	Leiter
Berichte		
AP5.1	Technischer Bericht	ALL
AP5.2	Management Summary	FLO
AP5.3	Kurzfassung / Abstract	WIM
AP5.4	Persönliche Berichte	ALL
Referenzen		
UAP5.5	Dokumentenplan	FLO
UAP5.6	Glossar	WIM
UAP5.7	Literaturverzeichnis	WIM
Handbücher		
AP5.8	Installationsanleitung	FLO
AP5.9	Benutzerhandbuch	WIM
Abschluss		
AP5.10	Interner Abschluss	ALL
AP5.11	Abnahme	ALL

5. Phase Abschluss



Arbeitsumgebung

Arbeitsumgebung

Software

AP1.1

Es müssen verschiedenste Software-Tools für den Einsatz in dem Projekt eruiert und installiert werden

Arbeiten:

- Evaluation & Installation der verschiedenen Software-Tools
- Microsoft.NET Laufzeitumgebung Version 1.1
- DirectX 9.0 Software Development Kit SDK
- Visual Studio .NET 2003
- Office 2003 / Word, Excel, Outlook, Project, Visio
- CVS System, Clients & Server, installieren
- CVSNT
- WinCVS
- Rational Rose 2003
- XMLSpy 2004
- CodeCounter
- PolyStyle CodeBeautifier
- GhostScript
- Skype
- Dreamweaver MX2004

Zu bearbeitende Dokumente:

- Projekt Management Plan / Entwicklungsprozess

Templates

AP1.2

Für die Dokumentation werden Vorlagen erstellt, um ein einheitliches Erscheinen zu garantieren. Es wird fortlaufend an der Dokumentation gearbeitet.

Arbeitsschritte:

- Templates für die Dokumentation und die Protokolle des Projektes erstellen

Zu bearbeitende Dokumente:

- Template Dokumente
- Template Sitzungsprotokolle



Projektmanagement

Projektmanagement

Aufgabenstellung

AP1.3

Dieses Dokument gibt eine Übersicht der verlangten Anforderungen an das Projekt. Hier werden die einzelnen Aufgaben sowie deren Teilaufgaben definiert.

Im Vorfeld müssen Informationen und Unterlagen zur Projektaufgabe und zu den zu verwendenden Technologien beschafft werden. Ausserdem müssen administrative Arbeiten durchgeführt werden.

Die Ausgangslage soll analysiert und daraus resultierend die Aufgabenstellung dokumentiert werden. Anschliessend kann das Projektziel gesteckt und der Arbeitsumfang abgeschätzt werden.

Arbeitsschritte:

- Kick Off Meeting
- Projektziele eruieren
- Projektumfeld studieren
- Unklarheiten bereinigen

Zu bearbeitende Dokumente:

- Aufgabenstellung

Projekt Homepage

UAP1.4

Zur Unterstützung des Projektablaufes und der Informationsbereitstellung wird eine Projekt Homepage erstellt, auf der alle News, Informationen sowie alle erstellten Dokumente einsehbar sind.

Arbeitsschritte:

- Homepage programmieren
- Inhalte definieren und einfügen

Resultate:

- Projekt Homepage



Projekt Management Plan

AP1.5

Dieses Dokument regelt organisatorische Belange. Die Organisationsstruktur, die Ziele und die Entwicklungsmethoden sind festgehalten. Es dient als Arbeits- bzw. Richtungsvorgabe für die Mitarbeiter. Ausserdem eröffnet es dem Kunden den Einblick in die Arbeitsweise des Entwicklerteams.

Der Projekt Management Plan ist ein Arbeitspaket, das sich über die gesamte Dauer der Diplomarbeit erstreckt. Der Schwerpunkt liegt bei diesem Paket in den Anfängen des Projektes. Es wird jedoch während dem gesamten Verlauf weiter bearbeitet.

Arbeitsschritte:

- Allgemeine Informationen zusammentragen und festhalten
- Projektorganisation eruieren
- Projektmanagement definieren
- Risiken analysieren und beurteilen
- Entwicklungsprozess und –plan aufstellen

Zu bearbeitende Dokumente:

- Projekt Management Plan

Projektplan / Zeiterfassung

UAP1.6

Der Projektplan ermöglicht einen Überblick über das gesamte Projekt. Tätigkeiten, Zeit und Ressourcen werden hier geplant und eingetragen.

Arbeitsschritte:

- Erarbeiten der Tätigkeiten
- Bestimmen der Reihenfolge der Tätigkeiten
- Zeitschätzungen für die einzelnen Tätigkeiten

Zu bearbeitende Dokumente:

- Projektplan



Style Guide

AP1.7

Dieses Dokument beschreibt den Code Style Guide der Diplomarbeit. Dieser Code Style Guide bestimmt die grundlegenden Programmierregeln und das Aussehen des Source Codes der zu entwickelnden Applikation.

Arbeitsschritte:

- Erarbeiten der Art und Weise des Programmierens

Zu bearbeitende Dokumente:

- Style Guide

Protokolle

UAP1.8

Als Vorbereitung und am Ende einer Sitzung wird ein Protokoll erstellt, in dem erfasst wird, wer beteiligt gewesen war, was besprochen wurde und wer danach was zu tun hat (Aufgabenliste). Bei Erreichung eines Meilensteins wird erfasst, wie der aktuelle Projektstand ist, was noch zu tun wäre und das weitere Vorgehen festgehalten.

Arbeitsschritte:

- Vor- und Nachbereitung des Protokolls

Zu bearbeitende Dokumente:

- Sitzungsprotokoll
- Meilensteinprotokoll

Sitzungen

UAP1.9

Während des ganzen Projektablaufes gibt es verschiedene Sitzungen: Sitzungen mit dem Betreuer / Auftragsgeber, Reviews, Ad-Hoc-Meetings und sonstige Besprechungen.

Arbeitsschritte:

- Sitzung durchführen



Analyse

Analyse

Anforderungsspezifikation

AP2.1

Beginnend mit einer allgemeinen Beschreibung der zu entwickelnden Applikation, beschreibt dieses Dokument die Benutzergruppen, die zu erwartenden Probleme und verschiedene Annahmen und Abhängigkeiten. Es folgen detaillierte Beschreibungen der einzelnen Anforderungen sowie Randbedingungen für den Entwurf.

Arbeitsschritte:

- Anforderungen an die Prototypen
- Anforderungen an die Applikation

Zu bearbeitende Dokumente:

- Anforderungsspezifikation

Analyse

AP2.2

Dieses Dokument dient dazu, aufzuzeigen, warum und wieso einzelne Implementierungen von Teilanforderungen in bestimmter Weise gelöst wurden.

Arbeitsschritte:

- In der ersten Iteration: Analyse der Prototypen
- In der zweiten Iteration: Analyse der Applikation aufgrund der gewonnenen Erkenntnisse aus den Prototypen

Zu bearbeitende Dokumente:

- Analyse



Design

Design

Design

AP3.1

Diese Dokumente beschreiben die Architektur und das Design der evolutionären Prototypen der 1. Iteration sowie die zu entwickelnden Applikation der 2. Iteration. Sie ermöglichen einem Entwickler den Einstieg in den Programmcode zu vereinfachen, damit dieser die Applikation erweitern kann. Des Weiteren ist daraus ersichtlich, warum ein solches Design und eine solche Architektur verwendet wurden.

Ziel des Dokumentes ist es:

- einem Entwickler den Einstieg in den Programmcode zu erleichtern, damit dieser die Applikation erweitern kann.
- nachvollziehen zu können, warum ein solches Design und eine solche Architektur verwendet wurden.

Arbeitsschritte:

- In der ersten Iteration wird die Architektur der Prototypen beschrieben. Danach folgen die Klassendiagramme, einige Sequenz- und Kollaborationsdiagramme, die den dynamischen Aspekt der Anwendung dokumentieren
- In der zweiten Iteration wird das Design der Applikation entwickelt. Dies beinhaltet die Erklärung der Architektur, wiederum Klassen- und Sequenzdiagramme und zum Schluss noch das Design des User Interfaces.

Zu bearbeitende Dokumente:

- Design (Prototypen&Applikation)



Testspezifikationen (erst in der 2. Iteration zu erstellen) AP3.2

Dieses Dokument enthält die Testspezifikation, sowie die Testberichte für das Projekt. Folgendes wird damit festgelegt:

- Welche Funktionen getestet werden.
- Wie die Funktionen getestet werden.
- Ob die Anforderungen eingehalten wurden.
- Ob die Zuverlässigkeit und Qualität der Applikation Zellomat3D genügend gut ist.

Arbeitsschritte:

- Prioritätsanalyse
- Testfälle
- Die Prioritätsanalyse begründet, warum die einzelnen Funktionen verschieden ausführlich getestet werden. Damit wird festgelegt, wie detailliert getestet wird.
- Testen der Applikation
- Testberichte

Zu bearbeitende Dokumente:

- Testbericht



Implementation der 1. Iteration

Implementation 1. Iteration

Prototypen Modul Rohdatenberechnung

AP4.1

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen mehrerer evolutionärer Software-Prototypen um für die 2. Iteration die richtigen Entscheidungen treffen zu können.

Arbeitsschritte:

- Programmierung der verschiedenen Prototypen
- Testen der Prototypen auf ihre Leistungsfähigkeit und Performance sowie auf ihr Entwicklungspotential

Zu bearbeitende Dokumente:

- Analyse der 2. Iteration

Prototypen Modul Datenaufbereitung

AP4.2

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen mehrerer evolutionärer Software-Prototypen um für die 2. Iteration die richtigen Entscheidungen treffen zu können.

Arbeitsschritte:

- Programmierung der verschiedenen Prototypen
- Testen der Prototypen auf ihre Leistungsfähigkeit und Performance sowie auf ihr Entwicklungspotential

Zu bearbeitende Dokumente:

- Analyse der 2. Iteration

Prototypen Modul Visualisierung

AP4.3

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen mehrerer evolutionärer Software-Prototypen um für die 2. Iteration die richtigen Entscheidungen treffen zu können.

Arbeitsschritte:

- Programmierung der verschiedenen Prototypen
- Testen der Prototypen auf ihre Leistungsfähigkeit und Performance sowie auf ihr Entwicklungspotential

Zu bearbeitende Dokumente:

- Analyse der 2. Iteration



Implementation der 2. Iteration

Implementation 2. Iteration

Modul Rohdatenberechnung

AP4.4

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen des zu erstellenden Software-Elementes.

Arbeitsschritte:

- Programmierung der Software
- Testen des erstellten Source Codes

Zu bearbeitende Dokumente:

- Testprotokoll
- Testbericht

Modul Datenaufbereitung

AP4.5

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen des zu erstellenden Software-Elementes.

Arbeitsschritte:

- Programmierung der Software
- Testen des erstellten Source Codes

Zu bearbeitende Dokumente:

- Testprotokoll
- Testbericht

Modul Visualisierung

AP4.6

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen des zu erstellenden Software-Elementes.

Arbeitsschritte:

- Programmierung der Software
- Testen des erstellten Source Codes

Zu bearbeitende Dokumente:

- Testprotokoll
- Testbericht



Deployment Programm

AP4.7

Alle getesteten und lauffähigen Software-Module werden hier zu einer grossen Applikation zusammengefügt.

Arbeitsschritte:

- Deployment der kompletten Applikation
- Entwickeln eines User Interfaces für die Applikation
- Testen der ganzen Software (Use Cases durchspielen)

Zu bearbeitende Dokumente:

- Testsprotokoll
- Testbericht

Automatenentwicklung

AP4.8

Dieses Arbeitspaket beinhaltet die Entwicklung, Programmierung und das Testen verschiedener zellulärer Automaten.

Arbeitsschritte:

- Programmierung verschiedener zellulärer Automaten
- Die besten als „Presets“ zur Applikation hinzufügen



Berichte

Berichte

Technischer Bericht

AP5.1

Dieses Dokument stellt die Projektaufgabe in einen grösseren Zusammenhang und liefert eine Beschreibung der Problemstellungen, deren Lösungen und Resultate.

Arbeitsschritte:

- Der technische Bericht gibt eine Einführung in die Aufgaben- und Problemstellung, beschreibt die Ergebnisse, die erzielt wurden und die Schlussfolgerungen zum Projekt.

Zu bearbeitende Dokumente:

- Technischer Bericht

Management Summary

AP5.2

Dieses Dokument dient dem Management, um einen Einblick in die Arbeit zu erlangen. Damit kann festgestellt werden, welche Ziele erreicht wurden und inwiefern diese Arbeit für das Unternehmen einen Nutzen bringen kann.

Arbeitsschritte:

- Das Management Summary beschreibt zu Beginn die Ausgangslage bezüglich des Projektes. Danach werden das Vorgehen und die erreichten Ziele erläutert. Zum Schluss gibt der Bericht einen Ausblick bezüglich der Risiken und der weiteren Massnahmen.

Zu bearbeitende Dokumente:

- Management Summary



Kurzfassungen / Abstract

AP5.3

Dieses Dokument gibt eine kurze Übersicht der gestellten Aufgaben, dem Ziel, der Umgebung sowie allgemeine Informationen über das Projekt.

Arbeitsschritte:

- Aufgabenstellung auswerten
- Kurzfassung der Studienarbeit schreiben (für die Projektdokumentation)
- Abstract der Studienarbeit erstellen (für die HSR)

Zu bearbeitende Dokumente:

- Kurzfassung
- Abstract

Persönliche Berichte

AP5.4

Zum Abschluss des Projektes werden alle Teammitglieder einen Überblick über ihre persönlichen Erfahrungen und Erlebnisse niederschreiben.

Arbeitsschritte:

- Notieren der Erfahrungen und Erlebnisse

Zu bearbeitende Dokumente:

- Persönliche Berichte



Referenzen

Referenzen

Dokumentenplan

UAP5.5

Dieses Dokument enthält eine Übersicht über alle erstellten Dokumente während des gesamten Projektablaufes. Die Verantwortlichkeiten, Versionisierung usw. werden hier für die einzelnen Dokumente beschrieben.

Arbeitsschritte:

- Alle zu erstellenden Dokumente erfassen
- Verantwortlichkeiten usw. definieren

Resultate:

- Dokumentenplan

Glossar

UAP5.6

Dieses Dokument erklärt alle speziellen Begriffe sowie alle gebrauchten Abkürzungen des Projektes.

Arbeitsschritte:

- Während und zum Schluss des Projektverlaufes sind die die speziellen Begriffe sowie alle gebrauchten Abkürzungen in das Glossar einzutragen

Zu bearbeitende Dokumente:

- Glossar

Literaturverzeichnis

UAP5.7

In diesem Dokument sind alle Quellen, die für die Diplomarbeit verwendet worden sind, aufgeführt.

Arbeitsschritte:

- Während und zum Schluss des Projektverlaufes sind die verwendeten Quelle in das Literaturverzeichnis einzutragen

Zu bearbeitende Dokumente:

- Literaturverzeichnis



Handbücher

Handbücher

Installationsanleitung

AP5.8

Dieses Dokument beschreibt die notwendigen Schritte um die Applikation zu installieren und diese ordnungsgemäss zu betreiben.

Arbeitsschritte:

- Einmal die Software installieren und dabei die einzelnen Arbeitsschritte festhalten. Diese anschliessend verständlich zu dokumentieren.

Zu bearbeitende Dokumente:

- Benutzerhandbuch

Benutzerhandbuch

AP5.9

Dieses Dokument beschreibt die Bedienung der Applikation während ihrer Laufzeit.

Arbeitsschritte:

- Alle Bedienungselemente erklären
- Tätigkeitsabfolgen zur Verwirklichung von Arbeitsabläufen oder gewünschte Tätigkeiten (Use Cases) verständlich dokumentieren

Zu bearbeitende Dokumente:

- Benutzerhandbuch



Abschluss

Abschluss

Interner Abschluss

AP5.10

Die gesamte Dokumentation und alle zugehörigen Elemente, Source Code usw. werden nochmals komplett überarbeitet, korrigiert und verbessert.

Voraussetzungen:

- Softwarecodierung abgeschlossen
- gesamte Dokumentation vorhanden und inhaltlich korrekt

Arbeitsschritte:

- Gesamte Dokumentation überarbeiten, auf Fehler prüfen und korrigieren
- Druck und Binden der nötigen Exemplare
- Daten für CD aufbereiten
- CD brennen und Beschriftung anbringen

Resultate:

- gesamte Dokumentation abgeschlossen, ausgedruckt und gebunden
- Datenträger (Projekt-CD inklusive Projekt-Homepage) mit allen Projektrelevanten Daten angefertigt

Zu bearbeitende Dokumente:

- gesamte Dokumentation und alle zugehörigen Elemente

Abnahme

AP5.11

Dies ist das abschliessende Arbeitspaket. Es wird beim letzten Treffen mit dem Auftragsgeber durchgeführt.

Arbeitsschritte:

- Übergabe der Dokumentation und des Datenträgers mit allen projektrelevanten Daten
- Erklärungen zur Software
- Wissenstransfer

Resultate:

- Herr Glatz hat eine Software, mit der die gewünschten Aufgaben (siehe Anforderungsspezifikation) durchgeführt werden können sowie einen Einblick in die Funktionsweise der Software erhalten.



Terminplan

Datum	Meilenstein	Produkte	Inhalt	Meilensteine
14.03.2005	Projektstart & Kick off Meeting	Aufgabenstellung	Besprechung der Aufgaben	
16.03.2005	Meilenstein Phase 1 Vorbereitung	Projekt Management Vorbereitungen	Projekttablauf, Arbeitsumfeld	
18.03.2005	Meilenstein Phase 2 Iteration 1 Analyse	Anforderungsspezifikation, Analyse	Analyse & Anforderungen der/an evolutionäre/n Prototypen	
21.03.2005	Meilenstein Phase 3 Iteration 1 Design	Design der evolutionären Prototypen	Erstellen des Designs für die Prototypen	
04.04.2005	Meilenstein Phase 4 Iteration 1 Implementation	Evolutionäre Prototypen	Implementation & Präsentation	
06.04.2005	Meilenstein Phase 2 Iteration 2 Analyse	Anforderungsspezifikation Analyse	Analyse und Anforderungsspezifikation der Applikation Zellomat3D	
11.04.2005	Meilenstein Phase 3 Iteration 2 Design	Design	Design der Applikation Zellomat3D	
02.05.2005	Meilenstein Phase 4 Iteration 2 Implementation	Fertiger Zellomat3D	Präsentation der Applikation Zellomat3D	
06.05.2005	Meilenstein Phase 5 Abschluss	Programm, Code, Dokumentation	Präsentation der Diplomarbeit	

Tabelle 1: Meilensteine

Der detaillierte Projektplan ist in der Datei „Projektplan - 16032005.mpp“ zu finden. Das geplante wöchentliche Arbeitspensum pro Student beträgt durchschnittlich 66 Stunden. Bei Eintritt eines Risikos werden zusätzliche Stunden eingesetzt.



Zeitauswertung

Projektstunden pro Woche

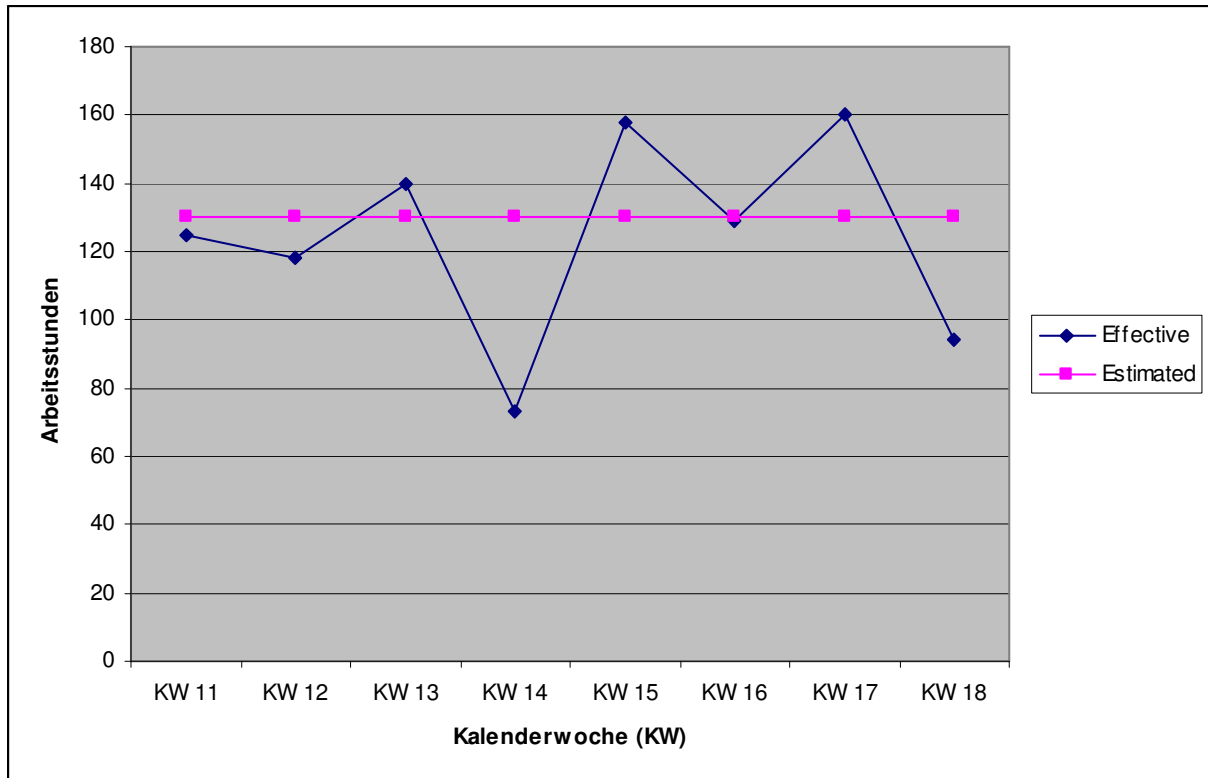


Abbildung 2: Projektstunden pro Woche

Wie das Diagramm in Abbildung 2 zeigt, wurde zwischen der dritten und vierten Woche des Projektes nicht sehr viel gearbeitet.. Grund dafür war die Analyse und das Design der Applikation, die weniger Zeit in Anspruch nahmen als prognostiziert und eingeplant war.

In der fünften bis achten Woche hingegen wurden die geplanten Zeiten überschritten, da die Implementation der Applikation mehr Zeit in Anspruch genommen hatte als dafür geplant war.

Am Schluss sackt die Grafik ab, weil Samstag und Sonntag der Kalenderwoche 18 nicht mehr in der Projektdauer liegen.

Der berechnete wöchentliche Durchschnitt über 8 Wochen hinweg liegt bei 62.6 Stunden pro Woche je Mitarbeiter.



Projektstunden aufsummiert pro Woche

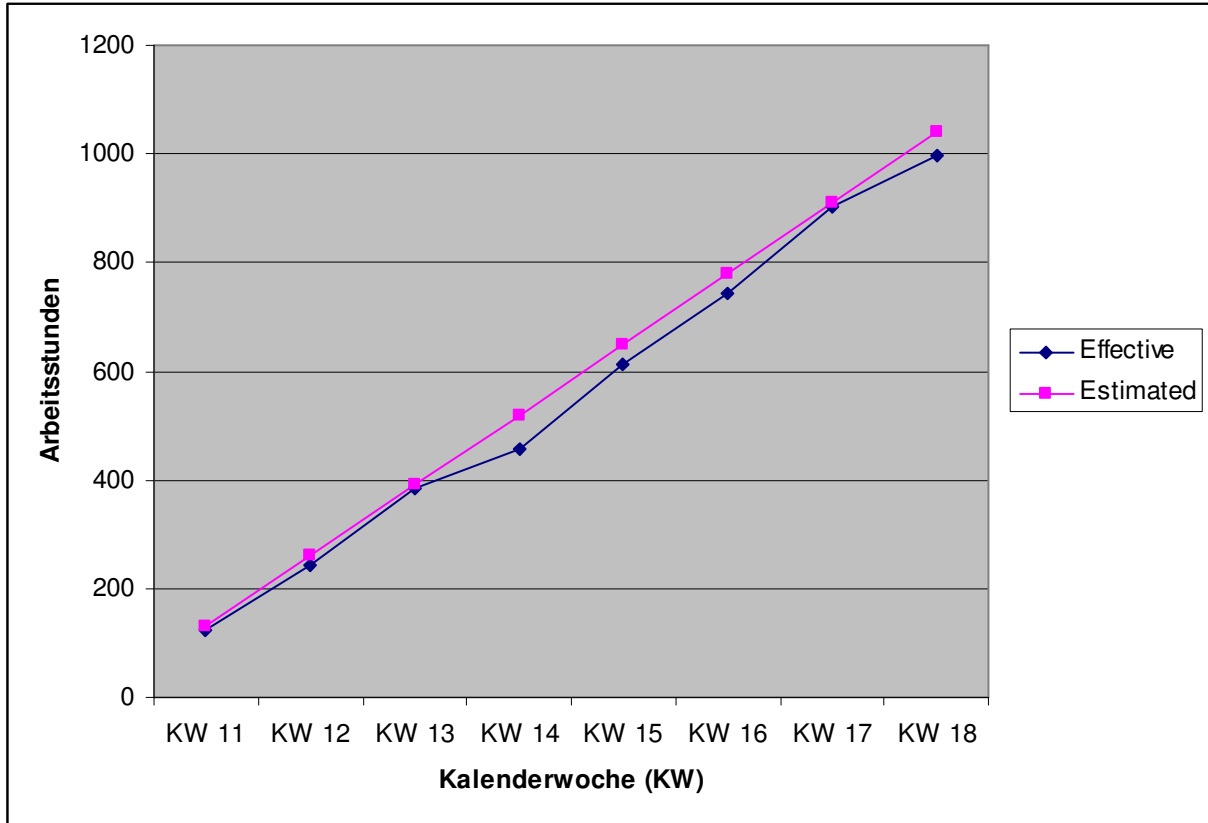


Abbildung 3: Projektstunden aufsummiert pro Woche

Die unter Punkt "Projektstunden pro Woche" beschriebenen Erscheinungen sind auch aus Abbildung 3 klar ersichtlich. Nach der zweiten Woche (KW 12) klappten die geplanten und die effektiven Stunden auseinander und fügten sich anschließend wieder zusammen



Projektstunden nach Phasen

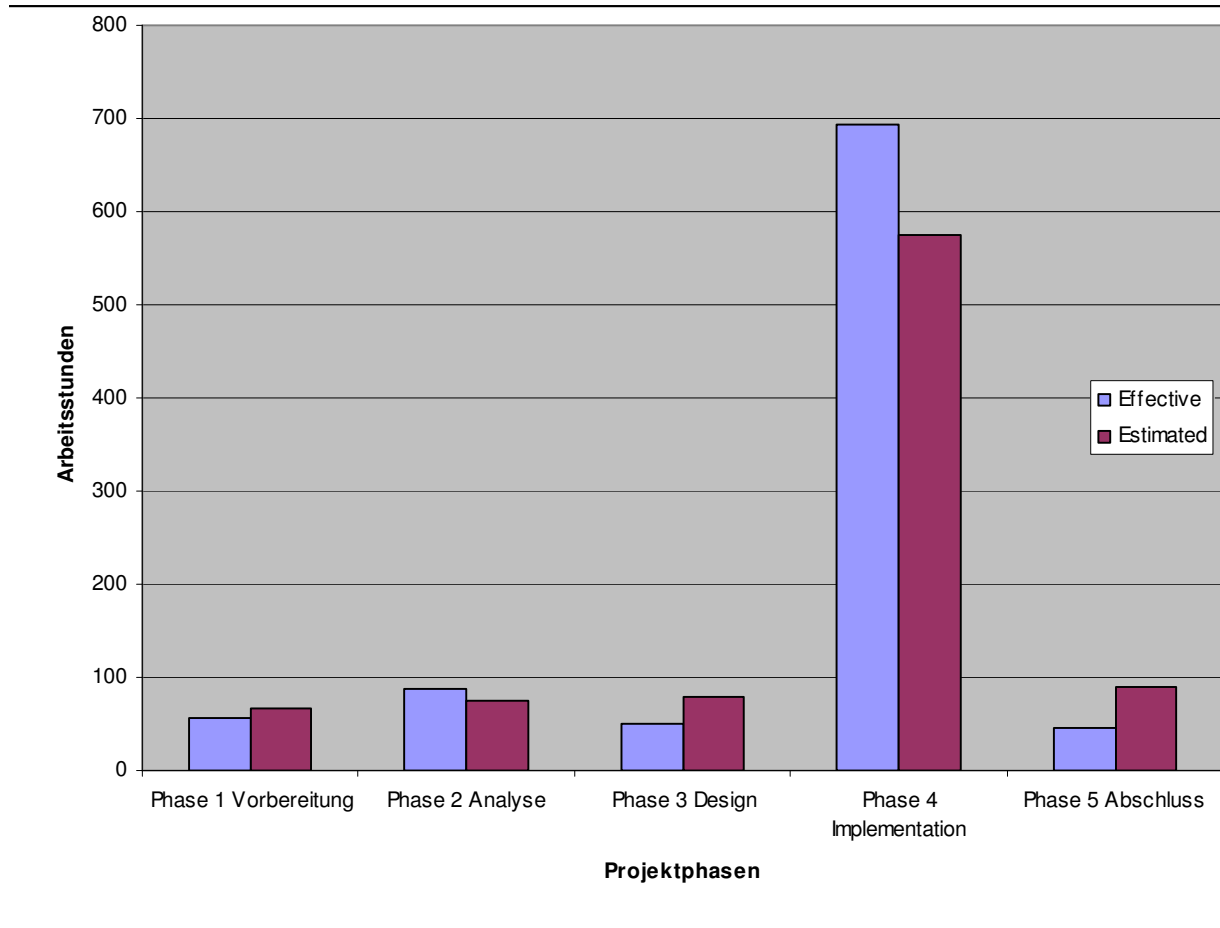


Abbildung 3: Projektstunden nach Phasen

Auffallend an dieser Grafik ist vor allem der Peak bei der Implementation. Der überhöhte Aufwand ist vor allem auf das Erlernen eines völlig unbekanntes Fachgebietes, nämlich der 3D Visualisierung mittels der Grafikkbibliothek DirectX 9.0, zurückzuführen. Des Weiteren wurde sehr viel Zeit in die Entwicklung von effektiven und schnellen Berechnungsalgorithmen für die zellulären Automaten gesteckt.



Teammitglieder Stunden pro Woche

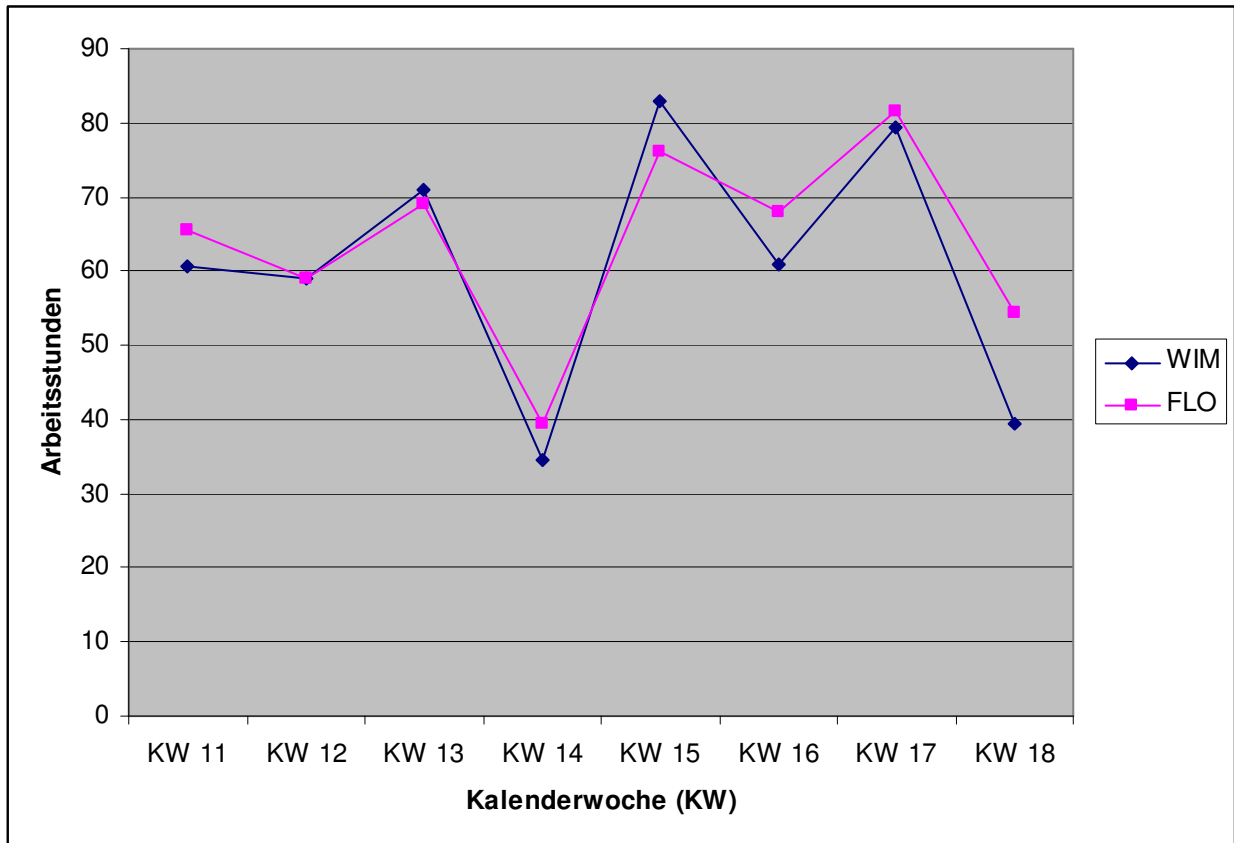


Abbildung 4: Teammitglieder Stunden pro Woche

Der Zeitaufwand von FLO und WIM ist beinahe identisch. Dies liegt sicher daran, dass wir meist gemeinsam zu Hause in unserer WG arbeiteten.

Der berechnete Durchschnitt pro Person liegt bei 61h/Woche für WIM und 64.1h/Woche für FLO.



**Teammitglieder
Stunden
aufsummiert**

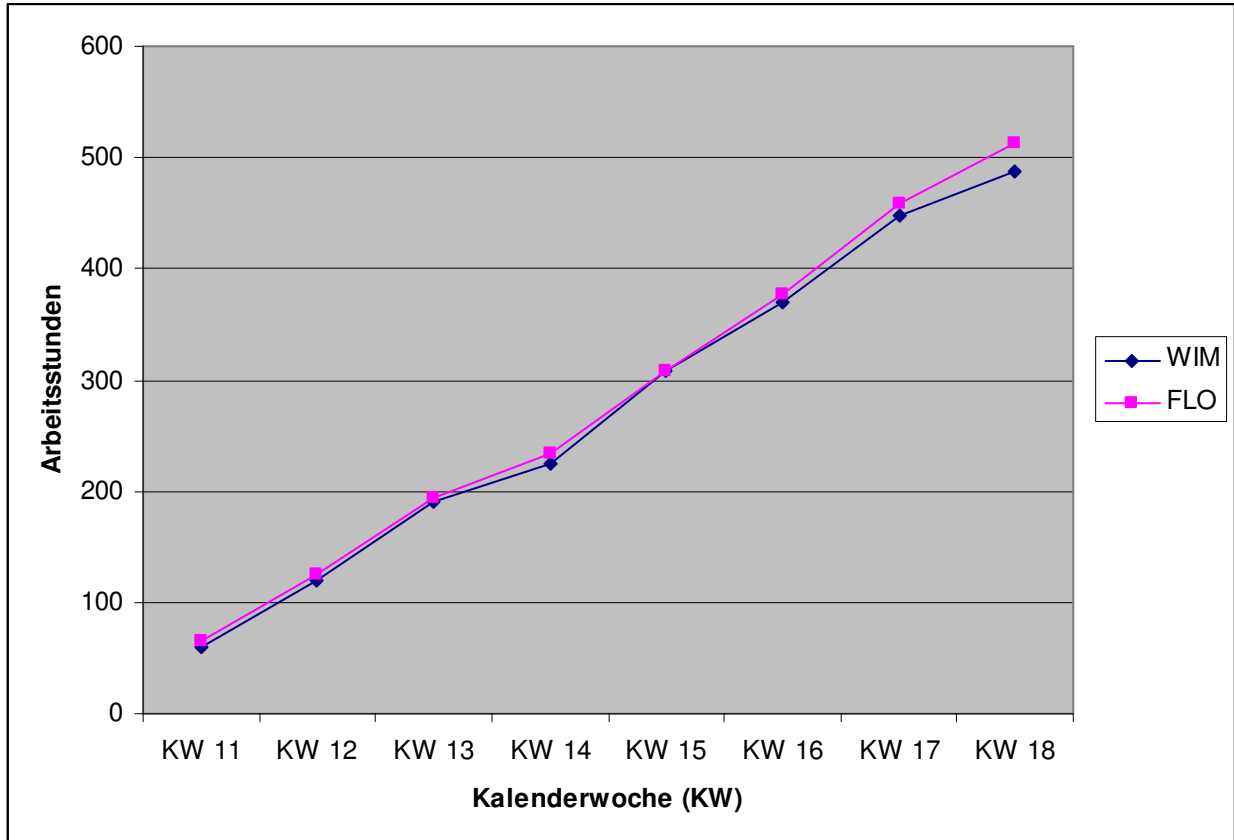


Abbildung 5: Teammitglieder Stunden aufsummiert

Betrachtet man die geleisteten Arbeitsstunden in aufsummierter Form, sind die Kurven von WIM und FLO beinahe identisch.



Zeit pro Tätigkeitsbereich

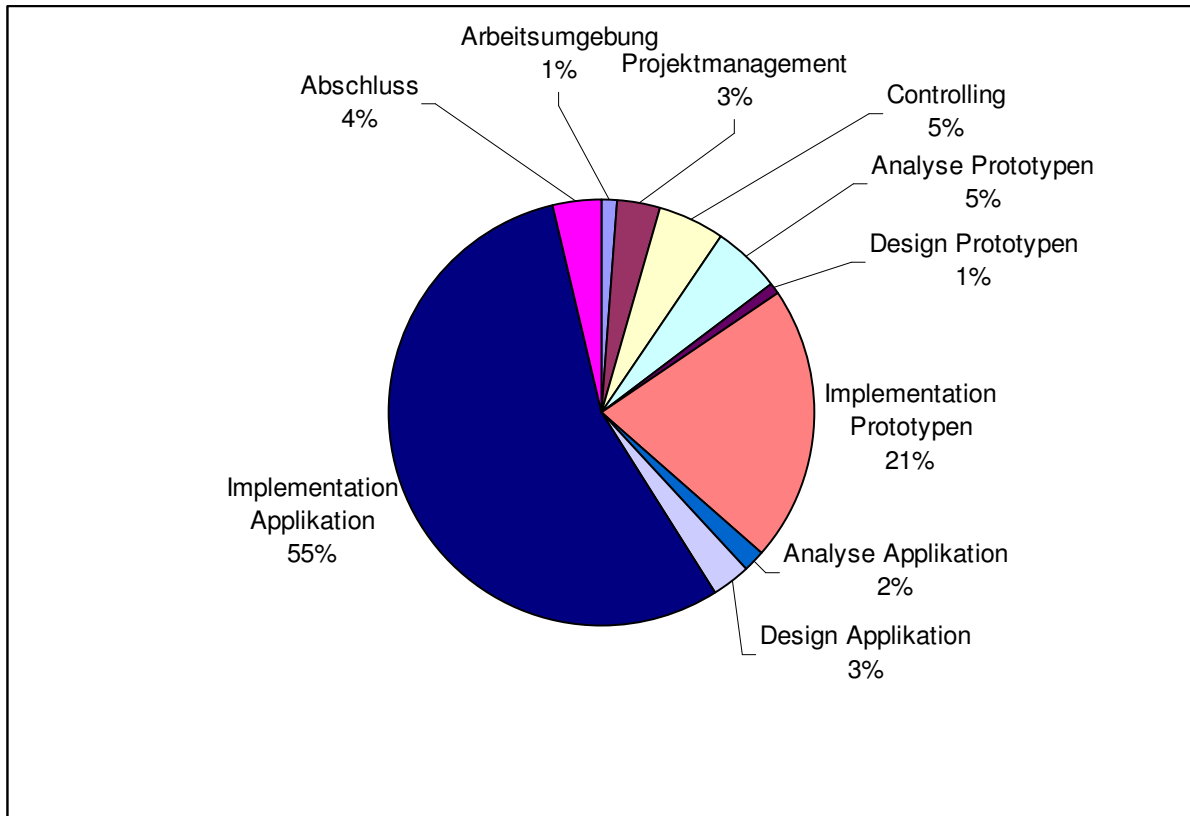


Abbildung 5: Zeit pro Tätigkeitsbereich

Hier ist klar ersichtlich, dass der grösste Teil (76%) der Projektdauer für die Realisierung der Prototypen sowie für die Implementation der Applikation gebraucht wurden. Die Analyse und das Design der zwei durchlaufenen Iterationen nahmen 11% und alles, was zur Unterstützung des Projektablaufes gebraucht wird, zu 13% an der Dauer beteiligt sind.



Zeit pro Phase

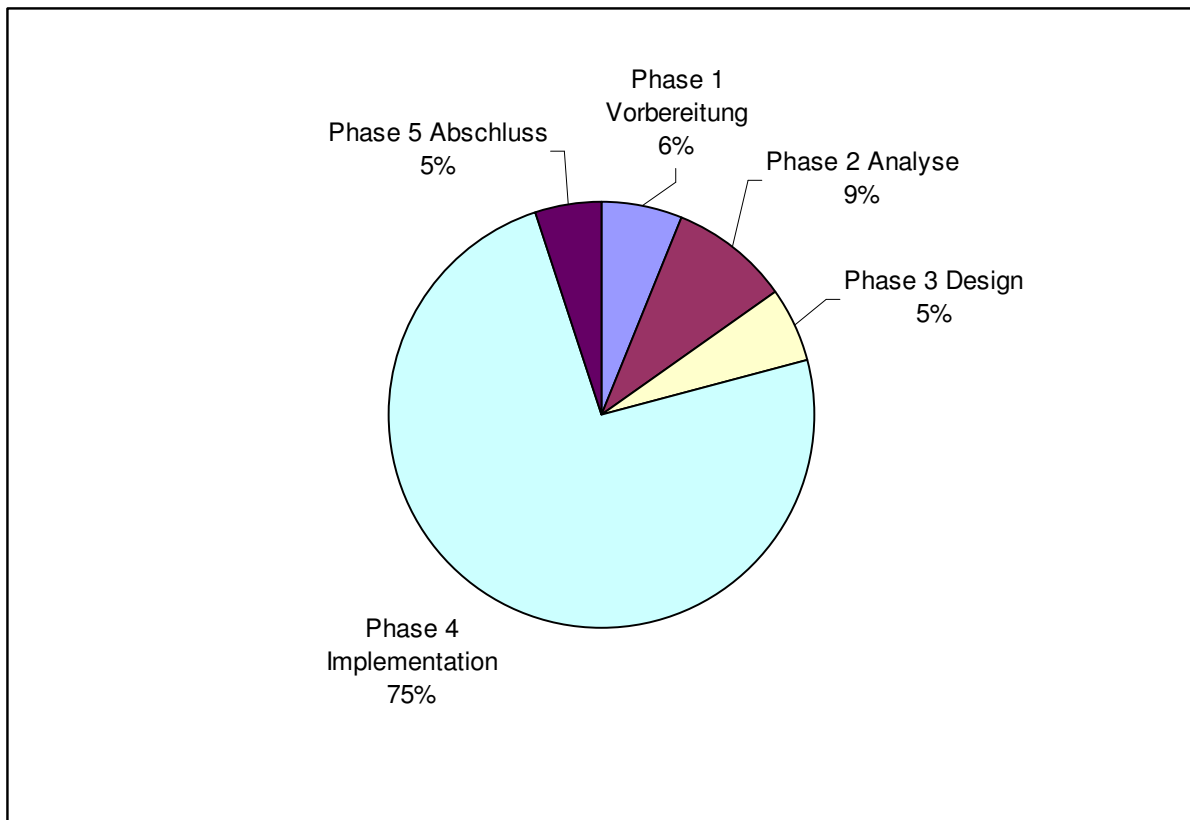


Abbildung 6: Zeit pro Phase

Fasst man die einzelnen Tätigkeiten nach Phasen zusammen, so fällt die Phase 1 mit 6% ins Gewicht. Die Phasen 2, 3 und 4 wurden in je zwei Iterationen durchlaufen. Die Phase 4 brauchte mit 75% mit Abstand am meisten Zeit.



Codestatistik

Tabelle 1 zeigt eine Übersicht der wichtigsten Kennzahlen des **Gesamtübersicht** Sourcecodes.

File Name	Total Lines	Code Lines	Comment Lines	Blank Lines
Eigencode:				
About	114	82	27	5
CellularAutomat	40	22	13	5
Controller	532	338	123	71
DataPreparation	357	243	70	44
FrameBuffer	148	84	44	20
PropertyReader	1000	787	56	157
Renderer	794	594	94	106
Rule	189	124	41	24
TableTree	1552	960	464	128
Visualization	1832	1214	396	222
Total Eigencode	6558	4448	1328	782
Fremdcode:				
Font	673	454	118	107
SettingsForm	1136	802	224	113
VisEnumerations	646	446	105	108
Total Fremdcode	2477	1702	447	328
Total	9035	6150	1775	1110

Tabelle1: Codestatistik Übersicht

Graphik
Zeilenstatistik

