



*SIMPL GmbH, Universitätsstraße 38, 70569 Stuttgart*  
Katharina Görlach,  
Peter Reimann,  
Mirko Sonntag  
Universitätsstraße 38  
70569 Stuttgart

Wolfgang Hüttig  
Projektleiter SIMPL  
Universitätsstraße 38  
70569 Stuttgart

## Angebot

17. Juli 2009

Sehr geehrte Frau Görlach, sehr geehrter Herr Reimann, sehr geehrter Herr Sonntag,

vielen Dank für Ihr Interesse an einem Angebot unserer Firma. Gerne würden wir ihre Vorstellungen im Rahmen eines Entwicklungsprojekts umsetzen.

Sie finden hierzu auf den folgenden Seiten unser ausführliches Angebot. Über eine Zusammenarbeit in diesem Projekt würden wir uns sehr freuen.

Bei Rückfragen können Sie sich selbstverständlich gerne jederzeit mit uns in Verbindung setzen.

Mit freundlichen Grüßen

Wolfgang Hüttig

## Angebot

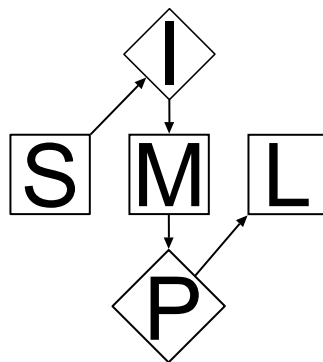
Version 1.0

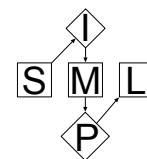
17. Juli 2009

Verfasser:

Daniel Brüderle, Firas Zoabi, Michael Hahn

---





# Inhaltsverzeichnis

<b>1</b>	<b>Ausgangssituation</b>	<b>5</b>
1.1	Projektbeteiligte . . . . .	5
1.2	Projektbeschreibung . . . . .	6
1.3	Entwicklungsgrundlagen . . . . .	6
<b>2</b>	<b>Projektaufgabe</b>	<b>7</b>
<b>3</b>	<b>Projektteam-Organisation</b>	<b>7</b>
3.1	Rollen im Team . . . . .	8
3.2	Kontaktdaten der Projektteam-Mitglieder . . . . .	9
<b>4</b>	<b>Nutzen</b>	<b>10</b>
<b>5</b>	<b>Leistungen im Rahmen des Angebots</b>	<b>10</b>
5.1	Auszuliefernde Software-Einheiten . . . . .	10
5.2	Software-Einheiten, die nicht zum Lieferumfang gehören . . . . .	11
<b>6</b>	<b>Terminplanung</b>	<b>11</b>
6.1	Beschreibung der Phasen . . . . .	12
6.1.1	Analyse . . . . .	12
6.1.2	Einarbeitung . . . . .	12
6.1.3	Spezifikation . . . . .	13
6.1.4	Entwurf . . . . .	13
6.1.5	Implementierung . . . . .	13
6.1.6	Test . . . . .	13
6.1.7	Abnahmetest . . . . .	14
6.1.8	Puffer . . . . .	14



<b>7</b>	<b>Unsere Konditionen</b>	<b>14</b>
7.1	Risiken . . . . .	14
7.2	Kostenschätzung . . . . .	15
7.3	. . . . .	16



# 1 Ausgangssituation

Zur Modellierung und Ausführung von Business Workflows wird die Sprache BPEL (Business Process Execution Language) bereits lange und effektiv eingesetzt. Die Stärken von BPEL, die, wie der Name schon sagt, im Bereich der Geschäftsprozesse liegen, sollen nun auch für wissenschaftliche Workflows ausgenutzt werden können. Im wissenschaftlichen Bereich gelten andere Anforderungen und dafür ist BPEL bis jetzt noch nicht geeignet. Da aber auch im wissenschaftlichen Bereich immer öfter Workflows eingesetzt werden, sei es bei der Durchführung von komplexen Berechnungen im Rahmen von Simulationen oder bei der Ausführung von Experimenten, sollen mithilfe einer Anpassung von BPEL die wissenschaftlichen Bedürfnisse an BPEL erfüllt werden. Kennzeichnet sind wissenschaftliche Workflows in der Regel durch große Datenmengen, die Heterogenität der Daten und somit den Bedarf nach großer Rechenkapazität. Diesen Anforderungen soll durch die neu zu entwickelnde Software Rechnung getragen werden.

## 1.1 Projektbeteiligte

In diesem Abschnitt geht es um die beteiligten Parteien des Projekts. Diese sind, nachfolgend als die Kunden bezeichnet, Katharina Görlach, Peter Reimann und Mirko Sonntag, sowie die Firma SIMPL, nachfolgend als Auftragnehmer bezeichnet, die durch Wolfgang Hüttig vertreten wird. Der Auftragnehmer besteht aus den Projektmitarbeitern: Wolfgang Hüttig, Michael Schneidt, René Rehn, Michael Hahn, Firas Zoabi, Daniel Brüderle und Xi Tu, wobei Wolfgang Hüttig der zentrale Ansprechpartner des Auftragnehmers für das Projekt ist. Im Rahmen dieses Angebots kommt dabei ein Vertrag zwischen den Kunden und dem Auftragnehmer zustande.

Die Kontaktdaten der Kunden sind wie folgt:

### **Dipl.-Inf. Katharina Görlach**

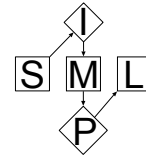
Institut für Architektur von Anwendungssystemen

Universitätsstraße 38, 70569 Stuttgart (Zimmer 1.328)

Telefon: +49 (0)711 7816-333

Fax: +49 (0)711 7816-472

E-Mail: [katharina.goerlach\(@\)iaas.uni-stuttgart.de](mailto:katharina.goerlach(@)iaas.uni-stuttgart.de)



## **Dipl.-Inf. Peter Reimann**

Institut für Parallele und Verteilte Systeme

Universitätsstraße 38, 70569 Stuttgart (Zimmer 2.467)

Telefon: +49 (0)711 7816-445

Fax: +49 (0)711 7816-424

E-Mail: Peter.Reimann(@)ipvs.uni-stuttgart.de

## **Dipl.-Inf. Mirko Sonntag**

Institut für Architektur von Anwendungssystemen

Universitätsstraße 38, 70569 Stuttgart (Zimmer 1.332)

Telefon: +49 (0)711 7816-202

Fax: +49 (0)711 7816-472

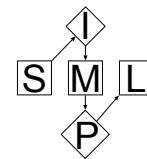
E-Mail: mirko.sonntag(@)iaas.uni-stuttgart.de

## **1.2 Projektbeschreibung**

Das Projekt wird als Time & Material Projekt durchgeführt und die Aufwände und Fortschritte des Projekts werden in regelmäßigen Abständen dem Kunden mitgeteilt. Dies erfolgt spätestens nach dem Abschluss einer Phase. Bei wichtigen Phasen auch innerhalb der Phasen, wenn es für sinnvoll erachtet wird.

## **1.3 Entwicklungsgrundlagen**

Die Entwicklung von SIMPL erfolgt nach erprobten softwaretechnischen Prinzipien. Dadurch wird eine hohe Qualität des Produkts gewährleistet. Wichtige Qualitäten sind dabei die Wartbarkeit und die Erweiterbarkeit des Systems, da es über einen längeren Zeitraum bestehen soll und in Zukunft um die Anbindung weiterer Datenquellen, Konzepte für den Datenzugriff und den Umgang mit weiteren Datenformaten ergänzt werden soll. Eine weitere wichtige Qualität ist die Skalierbarkeit des Systems, die eine sehr flexible Infrastruktur erlauben muss, da die Computersysteme, auf denen SIMPL später ausgeführt wird, in ihrer Leistung sehr weit auseinander gehen, d.h. vom normalen Desktop-Computer bis zum Supercomputer kann und soll alles möglich sein.



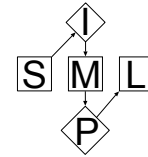
Robustheit und Usability der Software sind für uns innerhalb des Entwicklungsprozesses von großem Interesse. Die Usability sollte sich vor allem an unerfahrene und Nutzer mit wenig Kenntnissen im Umgang mit Workflows und BPEL richten und dafür die bestmögliche Transparenz liefern. d.h. dass die interne Prozesslogik der Software bestmöglich vom Benutzer abgeschirmt wird und dass er eine möglichst einfache und schnell verständliche Schnittstelle zur Software erhält, um die Verwendung von SIMPL für alle Benutzergruppen zu ermöglichen.

## 2 Projektaufgabe

Das Entwicklungsteam soll ein erweiterbares, generisches Rahmenwerk für die Modellierung und Ausführung von Workflows erstellen, welches den Zugriff auf nahezu beliebige Datenquellen ermöglichen soll. Bei den Datenquellen kann es sich beispielsweise um Sensornetze, Datenbanken und Dateisysteme handeln. Der Schwerpunkt soll klar auf wissenschaftlichen Workflows beruhen. Über das Rahmenwerk sollen beliebige Datenmanagement-Funktionen in einen BPEL-Prozess eingebunden werden können. Dafür werden bereits vorhandene Konzepte evaluiert und falls nötig erweitert oder angepasst. Es wird untersucht, inwiefern die Sprache BPEL ebenfalls erweitert werden muss. Für eine möglichst hohe Flexibilität soll ein dynamischer Ansatz gewählt werden, so dass erst während der Laufzeit des Systems die Datenquellen festgelegt werden können. Nichtsdestotrotz sollte auch die Möglichkeit bestehen, die Datenquellen statisch anbinden zu können. Eine Anforderung des Kunden ist, dass eine vorhandene BPEL-Engine sowie ein vorhandenes Modellierungstool um diese gewünschten Funktionen erweitert bzw. angepasst werden. Die BPEL-Prozesse sollen mit dem entsprechenden Modellierungstool spezifiziert und mit der BPEL-Engine ausgeführt werden können.

## 3 Projektteam-Organisation

In diesem Abschnitt wird die Organisation des Projektteams vorgestellt. Alle Teammitglieder haben dabei mindestens eine bestimmte Rolle, die mit entsprechenden Aufgaben und Pflichten verknüpft ist. Nachfolgend werden diese Rollen beschrieben und die Teammitglieder, die diese Rolle besitzen, angegeben. Danach folgen noch die Kontaktdaten aller Teammitglieder, damit der Kunde bei spezifischen Fragen direkt mit dem Verantwortlichen oder bei nichterreichbaren des Projektleiters mit dessen Stellvertreter Kontakt aufnehmen kann.



### 3.1 Rollen im Team

#### **Projektleiter: Wolfgang Hüttig (Stellvertreter: Michael Hahn)**

Der Projektleiter ist der zentrale Ansprechpartner für den Kunden. Er koordiniert sämtlichen Schriftverkehr und die Kommunikation zwischen den Kunden, den Betreuern und dem Team. Er ist auch an erster Stelle für das Projekt verantwortlich.

Seine zentralen Aufgaben liegen in der Verwaltung des Projekts, d.h. er plant den Ablauf des Projekts und der einzelnen Phasen, er bewertet und kontrolliert abschliessend, als letzte Instanz, alle erstellten Dokumente vor ihrer Freigabe und er kümmert sich um die Verteilung der einzelnen Arbeitspakete während des Projekts.

#### **Architekt: Michael Schneidt (Stellvertreter: Daniel Brüderle)**

Der Architekt ist zuständig für die Architektur des zu entwickelnden Softwaresystems und die Beschreibung der grundlegenden Komponenten und deren Zusammenspiel innerhalb des Softwaresystems. Er trifft die grundlegenden Entscheidungen über die Art und den Aufbau der Software-Architektur und überwacht deren Einhaltung während der Projektlaufzeit. Die Architektur soll die Einhaltung der unter Abschnitt ??? genannten Softwarequalitäten sicherstellen.

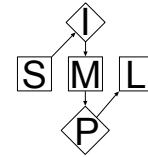
#### **Administrator: René Rehn (Stellvertreter: Michael Schneidt)**

Der Administrator ist in der Projektarbeit zuständig für die gesamte Infrastruktur auf Software- sowie auch auf Hardware-Ebene. Er ist für den Betrieb und die Wartung der Server und den Betrieb der darauf ausgeführten Software verantwortlich. Weiterhin sorgt er für die Bereitstellung der nötigen Software-Infrastruktur, d.h. er sorgt dafür, dass die zur Verfügung stehende Software, wie z.B. Werkzeuge für Entwicklung, Test und Entwurf, die zur Realisierung des Projekts benötigt wird, für das Projekt eingerichtet und betriebsbereit ist..

#### **Qualitätsmanagement: Michael Hahn, Firas Zoabi**

Der Qualitätsmanager sorgt dafür, dass alle Dokumente und auch Software-Komponenten die bestmögliche Qualität besitzen. Diese Qualität soll durch die Definition und Einhaltung von Richtlinien, durch die ständige Durchführung von Qualitätskontrollen und durch die Erstellung von einheitlichen Dokumentvorlagen erreicht werden. So soll während des gesamten Projektverlaufs die Qualität der Dokumente und Software-Einheiten maximiert werden.





**Entwickler: Michael Schneidt, René Rehn, Daniel Brüderle, Michael Hahn, Firas Zoabi, Xi Tu**

Als Entwickler arbeiten alle Teammitglieder ausser dem Projektleiter. Ein Entwickler arbeitet dabei im Rahmen der vom Projektleiter erteilten Aufgaben an Dokumenten und Software-Komponenten.

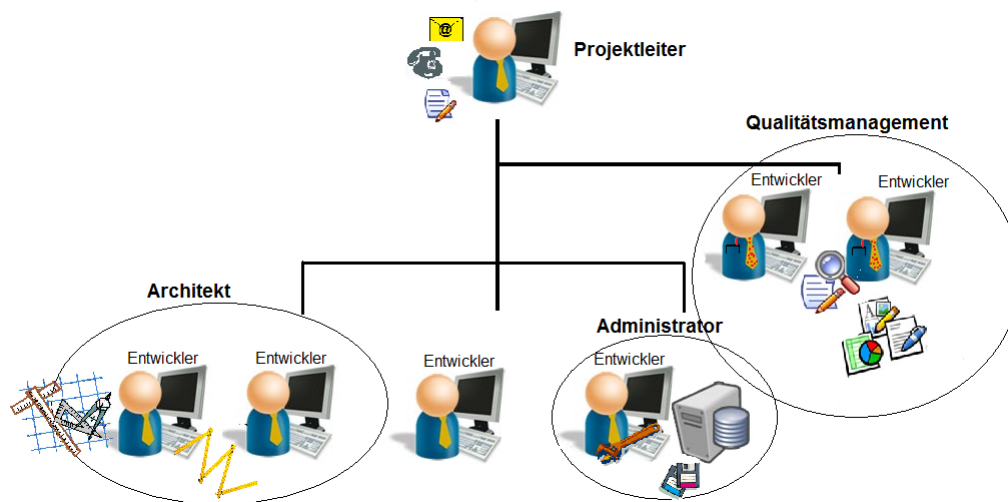


Abbildung 1: Rollen im Team

### 3.2 Kontaktdaten der Projektteam-Mitglieder

**Wolfgang Hüttig:** w.huettig(@)yahoo.de

**Daniel Brüderle:** daniel(@)brotherlee.de

**Michael Schneidt:** schneimi(@)studi.informatik.uni-stuttgart.de

**René Rehn:** Rene\_Rehn(@)gmx.de

**Firas Zoabi:** Firas.Zoabi(@)web.de

**Tu Xi:** tu\_xi(@)hotmail.com

**Michael Hahn:** hahnml(@)studi.informatik.uni-stuttgart.de



## 4 Nutzen

Die Arbeit im wissenschaftlichen Bereich und die Datenmengen, die damit verbunden sind, werden immer größer, heterogener und komplexer. Zur Vereinfachung und zur Automatisierung werden im Moment nur vereinzelt Scientific Workflows eingesetzt.

Durch das in diesem Projekt erstellte Rahmenwerk soll die Modellierung und die Ausführung von wissenschaftlichen Workflows mit BPEL, welches im Business-Bereich bereits viele Jahre erfolgreich eingesetzt wird, vereinfacht werden. Dazu soll das Rahmenwerk den Datenzugriff direkt aus einem Workflow heraus ermöglichen. . Dafür werden der Eclipse BPEL-Designer, Apache ODE und BPEL selbst erweitert, um so die direkte Anbindung von Datenquellen innerhalb von BPEL-Prozessen zu realisieren. Dadurch ist es auch für ungeübte Workflow-Benutzer wie z.B. Wissenschaftler und Ingenieure möglich, einfach und schnell wissenschaftliche Workflows zu erstellen und zu nutzen.

## 5 Leistungen im Rahmen des Angebots

### 5.1 Auszuliefernde Software-Einheiten

- Anforderungskatalog
- Projektplan
- Spezifikation
- Begriffslexikon
- Entwurf
- Handbuch
- Dokumentation der Erweiterungen
- Testfälle
- Testprotokolle
- Installationspaket, das alle benötigten Software-Komponenten beinhaltet
- 
- Quellcode

Alle Dokumente werden im PDF-Format und als L<sup>A</sup>T<sub>E</sub>X-Quelldateien geliefert.



## 5.2 Software-Einheiten, die nicht zum Lieferumfang gehören

Folgende Dokumente gehören nicht zum Lieferumfang:

- 
- 
- 
- Alle internen und organisatorischen Dokumente, wie z.B. Protokolle

## 6 Terminplanung

Das Studienprojekt läuft über zwei Semester und endet im Mai 2010. Jeder Projektteilnehmer hat dabei einen Arbeitsaufwand von ca. 480 Stunden zu leisten, dadurch entsteht ein Gesamtaufwand von ca. 3360 Stunden für alle 7 Teilnehmer. Die Projektphasen Analyse und Einarbeitung beanspruchen dabei 35%, Spezifikation und Entwurf (inkl. Fein-Entwurf) 24% und zuletzt die Codierung und der Test 33%. (siehe Abbildung 2)

Nach Berücksichtigung aller Einflussfaktoren von Seiten des Kunden und des Entwicklungsteams wird durch diese Planung vorausgesetzt, dass der Kunde das Endprodukt Mitte April 2010 testet und zwischen Anfang und Ende Mai 2010 nach Bedarf noch kleinere Änderungen durchgeführt werden.

Phase	Von / Bis	Aufwand
Analyse	13.05.2009-17.07.2009	658 h
Einarbeitung	17.07.2009-07.09.2009	526 h
Spezifikation	07.09.2009-31.10.2009	547 h
Entwurf	31.10.2009-24.12.2209	243 h
Implementierung	24.12.2009-12.03.2010	779 h
Test	12.03.2010-15.04.2010	334 h
Abnahmetest	15.04.2010	–
Puffer	15.04.2010-13.05.2010	273 h

Tabelle 1: Phasen des Projekts

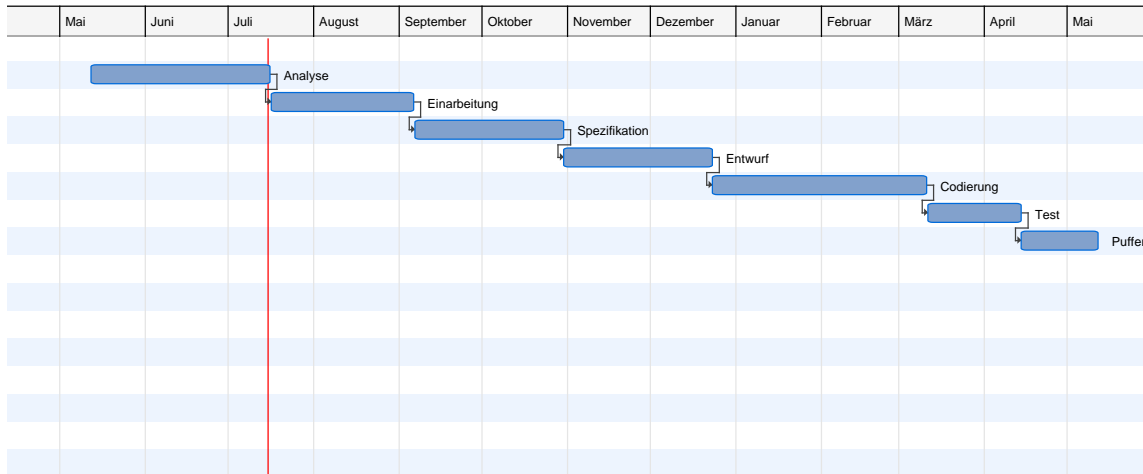
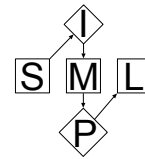


Abbildung 2: Terminplanung für die Entwicklungsphasen

## 6.1 Beschreibung der Phasen

In diesem Abschnitt wird jede der oben aufgeführten Phasen und die darin enthaltenen Aufgaben kurz näher erläutert.

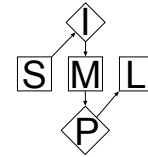
### 6.1.1 Analyse

Das Ziel der Analysephase ist es, durch Kundengespräche die funktionalen und nicht-funktionalen Anforderungen zu erheben, die der Kunde an das zu entwickelnde Softwaresystem stellt. Das Ziel ist es, einen Einblick in den momentanen Zustand des Arbeitsbereichs, für den die Software entwickelt werden soll, zu bekommen, um bereits realisierte Anforderungen aufzunehmen.

Am Ende der Analysephase wird aus den gewonnenen Erkenntnissen ein Anforderungskatalog erstellt, der als Referenz für die spätere Spezifikation der Software verwendet wird.

### 6.1.2 Einarbeitung

Da wenige bis gar keine Ansätze für Teilbereiche des Projekts wie z.B. die Verwendung von BPEL in Scientific Workflows existieren, müssen alle vorhandenen Ansätze evaluiert oder z.T. auch neue Ansätze geschaffen werden. Die Einarbeitungsphase dient dazu, sich in die verschiedenen Themenbereiche einzuarbeiten und festzustellen, in welchem Umfang Erweiterungsmaßnahmen notwendig sind.



### 6.1.3 Spezifikation

In dieser Phase wird die zu entwickelnde Software genau spezifiziert, d.h. in einem Dokument, der Spezifikation, werden alle funktionalen und nichtfunktionalen Anforderungen an die Software und ihre Schnittstellen präzise, vollständig und überprüfbar definiert. Ebenso werden spätere Abläufe im Umgang mit der Software und ihren Funktionen beschrieben. Die Spezifikation dient darüber hinaus als Referenz für viele nachfolgende Dokumente, wie z.B. das Handbuch und den Entwurf. Sie ist unerlässlich für die Testphase und die spätere Wartung oder für spätere Erweiterungen der Software. Anhand der Spezifikation kann sich auch der Kunde davon überzeugen, dass alle seine Anforderungen aufgenommen und nach seinen Vorstellungen spezifiziert wurden.

### 6.1.4 Entwurf

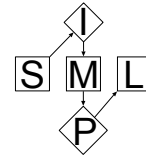
In der Entwurfsphase wird die Architektur der Software festgelegt und in verschiedenen Granularitäten in den Dokumenten Grob-Entwurf (Architektur des Gesamtsystems) und Fein-Entwurf (Architektur der einzelnen Module) festgehalten.

### 6.1.5 Implementierung

In der Implementierungsphase werden die Anforderungen aus der Spezifikation mit der Architektur des Entwurfs umgesetzt. Nach der Integration der implementierten Software-Module existiert am Ende dieser Phase eine ausführbare Software.

### 6.1.6 Test

Die Testphase enthält die drei Unterphasen Modultest, Integrationstest und Systemtest. Durch den Modultest werden die einzelnen Software-Module auf ihre korrekte Funktionalität innerhalb des Moduls getestet. Im Integrationstest wird die Integration der Module zum Gesamtsystem überprüft, d.h. es wird überprüft, ob alle Module korrekt zusammenarbeiten. Der Systemtest dient dazu die Software gegen die Spezifikation zu prüfen, d.h. alle Anforderungen aus der Spezifikation werden mit geeigneten Testfällen überprüft und die Software entsprechend korrigiert, falls sie nicht die gewünschten Ergebnisse liefert. Für alle drei Testarten gilt, dass immer eine Reihe von Testfällen definiert und deren Ausführung in einem Testprotokoll aufgezeichnet wird. Falls während eines Tests Fehler auftreten, so werden diese nach Ausführung des Test von den



Entwicklern korrigiert und dann erneut geprüft. Die in der Testphase erzeugten Dokumente, die Testfälle und Testprotokolle, sind auch für den Kunden wichtig, da sie als Referenz für seine internen Tests während der Abnahme genutzt werden können.

#### **6.1.7 Abnahmetest**

Am Ende der Testphase erfolgt die Abnahme der Projekts und aller genannten Dokumente (siehe Kapitel 5.1).

Dem Kunden wird damit Zeit zur Verfügung gestellt sicherzustellen, dass das von uns gelieferte Produkt seinen Anforderungen entspricht. Ist das der Fall, dann gilt das Projekt als abgeschlossen, andernfalls werden die nötigen Nachbesserungen oder Änderungen durchgeführt.

#### **6.1.8 Puffer**

Die Puffer-Phase ist ein extra Zeitfenster vor dem endgültigen Abgabetermin Ende Mai 2010. In dieser Phase können noch kleinere Änderungswünsche, die der Kunde nach dem Abnahmetest eventuell hat, umgesetzt werden.

## **7 Unsere Konditionen**

### **7.1 Risiken**

Aufgrund der geforderten sehr breiten Verwendungsmöglichkeiten im Rahmen der wissenschaftlichen Workflows besteht ein hohes Unsicherheitspotential im Bezug auf die Implementierungsdauer und auf sich ändernde Anforderungen.



Risiko	%	Kosten	Gegenmaßnahmen
Mitarbeiter fällt aus	20	2 MM	Wissen teilen, Zeitarbeit
Produkthaftung	5	1-10 MM	Versicherung abschliessen
Änderung der Anforderungen	50	1-5 MM	Flexible Struktur herstellen, regelmäßige Rücksprache mit Kunden
Qualität nicht ausreichend	30	3 MM	längere Testphase
Werkzeuge nicht einsetzbar	20	5 MM	Organisatorischer Slack schaffen
Infrastruktur Probleme	20	1 MM	einheitliche, zertifizierte Hardware
Anforderungen werden unterschätzt	40	5 MM	Schulungen
Software nicht lauffähig	20	6 MM	Prototyping
“Wildwuchs” der Änderungen	10	2 MM	Versionsmanagement
Entwicklungsverzug	30	6 MM	Überstunden
Kundenwünsche unrealistisch	30	1-5 MM	Anforderungen reduzieren.

## 7.2 Kostenschätzung

Unsere internen Berechnungen nach der Funktion-Point-Methode ergab folgende Kostenschätzung :

**Gesamtentwicklungszeit :**

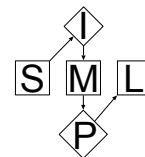
19 MM (Mann-Monate)

**Entwicklungszeit pro Mitarbeiter :**

19 MM / 7 Mann = 2,71 Monate Entwicklungszeit pro Mitarbeiter

**Zuschlag Entwicklungszeit (wg. Krankheit / Urlaub ) :**

2,71 Monate \* 1,10 = 2,981 Monate Entwicklungszeit pro Mitarbeiter



### **Mitarbeiterstunden :**

Die Mitarbeiter haben alle eine 5 Tage Woche mit jeweils 8 Stunden, können also im Monat 160 Stunden arbeiten.

Bei einer effektiven Entwicklungszeit von gerundet 3 Monaten ( 2,981) wird jede Person 480 Stunden arbeiten.

### **Stundensätze**

Entwickler : 69€ zzgl. MwSt.

Projektleiter : 99€ zzgl. MwSt.

### **Gesamtkostenschätzung :**

6 Entwickler x 69€/Stunde x 160 Stunden x 3 Monate = 198.720€

1 Projektleiter x 99€/Stunde x 160 Stunden x 3 Monate = 47.520€

**Gesamt : 246.240€ zzgl 19 % MwSt. = 293.025€**

### **7.3**