

# Java

# Thread Skalierung

# SPMP

## Software Project Management Plan

HTA Horw

**Änderungskontrolle**

<b>Version</b>	<b>Datum</b>	<b>Ausführende Stelle</b>	<b>Bemerkungen/Art der Änderung</b>
1.1	2006-10-16	Rainer Meier	Initial Release
1.2	2006-10-17	Rainer Meier	Tabellenbeschriftung „Links“
1.3	2006-10-17	Rainer Meier	Verantwortlichkeiten nach Rollen aufgeteilt, neues Tool: TortoiseCVS
1.4	2006-10-17	Rainer Meier	Zielkatalog
1.5	2006-10-17	Rainer Meier	Risikoportfolio, Projektplan, Meilensteine, Dokumentverantwortlichkeiten
1.6	2006-10-18	Marcel Aregger	Risiko-Mgmt, Arbeitspakete, Zeitplanung
1.7	2006-10-23	Marcel Aregger	Input Review Projektplanung/Zielkatalog
1.8	2006-10-31	Marcel Aregger	Dokument Evaluation und Umsetzung in SDD zusammengefasst, Kategorie Ziel #11
1.9	2006-14-11	Marcel Aregger	Umbenennung der Zielsetzungen {Z?}-Notation Abgabedatum Webabstrakt

**Prüfung und Freigabe**

<b>Vorname/Name</b>	<b>Dokumentversion</b>	<b>Status</b>	<b>Datum</b>	<b>Visum</b>
Rainer Meier	1.9	Final	2006-11-16	
Marcel Aregger	1.9	Final	2006-11-16	

## 1. Inhaltsverzeichnis

<b>1. Inhaltsverzeichnis .....</b>	<b>3</b>
<b>2. Einleitung .....</b>	<b>5</b>
2.1. Projektübersicht .....	5
2.1.1. Ausgangslage.....	5
2.1.2. Zielsetzungen .....	5
2.1.3. Projektumfang .....	5
2.2. Projektergebnisse .....	6
2.3. Aktualisierungswesen .....	6
2.4. Referenzierte Dokumente.....	6
2.5. Definitionen und Abkürzungen.....	7
2.6. Links.....	7
<b>3. Projektorganisation.....</b>	<b>8</b>
3.1. Prozessmodell .....	8
3.2. Organisationsstruktur.....	8
3.3. Organisatorische Abgrenzungen und Schnittstellen.....	8
3.4. Verantwortlichkeiten im Projekt .....	8
<b>4. Management Prozess.....</b>	<b>11</b>
4.1. Ziele und Prioritäten.....	11
4.1.1. Technische Zielsetzungen .....	11
4.1.2. Methodische Zielsetzungen .....	12
4.2. Annahmen und Abhängigkeiten.....	12
4.3. Risikomanagement .....	12
4.3.1. Massnahmen.....	12
4.3.2. Umsetzung und Wirksamkeit .....	13
4.3.3. Identifizierte Risiken .....	14
4.3.4. Risikolandschaft .....	15
4.4. Monitoring und Controlling .....	16
<b>5. Technischer Prozess.....</b>	<b>17</b>
5.1. Methoden, Tools, Techniken .....	17
5.2. Software Dokumentation.....	18
5.3. Projektunterstützung .....	18
<b>6. Arbeitspakete und Zeitplan .....</b>	<b>19</b>
6.1. Arbeitspakete .....	19
6.2. Abhängigkeiten .....	19
6.3. Benötigte Ressourcen.....	19
6.4. Zeitplan .....	20
6.4.1. Reviews .....	21

6.4.2. Meilensteine .....	21
6.4.3. Dokumente .....	22
<b>7. Verzeichnisse.....</b>	<b>23</b>
7.1. Tabellenverzeichnis .....	23
7.2. Abbildungsverzeichnis .....	23
7.3. Index .....	23

## 2. Einleitung

### 2.1. Projektübersicht

#### 2.1.1. Ausgangslage

Softwareskalierbarkeit wurde in der Vergangenheit ermöglicht durch die Verwendung von Technologien wie Pipelines oder Hyperthreading und der stetigen Erhöhung der Taktraten verwendeter Single-Core Prozessoren. Das Mooresche Gesetz hat in der Chip-Industrie weiterhin Gültigkeit (Verdopplung der Anzahl Transistoren alle 18 Monate), für die Entwickler neuer Chipserien ändern sich jedoch die Verwendungsmöglichkeiten dieser zusätzlichen Transistoren.

Die weitere Erhöhung der Taktrate neuer Chips mit der gleichen Geschwindigkeit wie in der Vergangenheit ist aufgrund der erzeugten Wärmeleistung nicht möglich und sinnvoll. Die zusätzlichen Transistoren können jedoch dazu verwendet werden, weitere Speicher mit geringer Latenz und mehrere Kerne auf einem Chip zu platzieren.

Multicore oder eben Multiprozessor-Architekturen nutzen aber wenig, wenn die verwendete Software nicht explizit für eine solchen Betrieb ausgelegt ist bzw. programmiert wurde. Faktisch gesehen vollzieht sich im Themenbereich der Skalierung von Applikationen ein Wandel von der Hardwareentwicklung hin zur gezielten Softwareentwicklung. Die Verantwortung skalierbare Applikationen zu entwickeln wird zusehends an den SW-Entwickler übertragen.

#### 2.1.2. Zielsetzungen

Übergeordnete Zielsetzung dieser Diplomarbeit (DA) ist das Thema der Skalierbarkeit von Anwendungen im einer Multi-Core bzw. Multi-CPU-Umgebung ganzheitlich zu beleuchten. Die vertiefte Analyse dieser Thematik soll hierbei auf Betriebssystemebene (primär Windows XP) und Code-Ebene (primär Java) erfolgen.

#### 2.1.3. Projektumfang

Die vorliegende DA wird von zwei Diplomanden durchgeführt. Der planbare Aufwand beträgt in Summe 420 Stunden die zwischen dem 16. Oktober 2006 und dem 24. November 2006 abgearbeitet werden. Methodisch betrachtet lässt sich die Aufgabenstellung in folgende Schwerpunkte gliedern:

**Tabelle 1 Projektschwerpunkte**

Schwerpunkt	Beschreibung
Projektmanagement	Planung, Controlling Projektverlauf
Basisanalyse	Erarbeitung Grundlagen HW und SW
Evaluation	Testplattform, Testkonzept und Tooling
Umsetzung/Implementierung	Testplattform, Implementierung Konzepte, Testing
Auswertung	Interpretation der Resultate, Empfehlungen
Dokumentation	Dokumentation der DA

## 2.2. Projektergebnisse

Die Resultate der Arbeit werden in einer Gesamt-Dokumentation festgehalten, die wiederum mehrere Teildokumente enthält:

**Tabelle 2 Projektergebnisse**

Dokumente	Beschreibung
SCMP	Configuration Management
SPMP	Projektmanagement
Projektplan	Ablaufplan, Arbeitspakete Projekt
Basisanalyse	Grundlagenanalyse
SDD	Evaluation; Definition von HW-Plattform/OS/VM und Testtools Umsetzung; Testumfang und Implementierung Konzepte
STD	Beschreibung Testkonzept und Testreihe
Conclusion	Interpretation der Resultate

Code oder Codefragmente sowie (freeware) Testtools werden ebenfalls aufbereitet bzw. abgegeben:

**Tabelle 3 Projektergebnisse 2**

Code	Beschreibung
Binaries, Source Code	SW-Implementierung
Tools	Verwendete (Freeware) Testtools

## 2.3. Aktualisierungswesen

Beim vorliegenden Dokument handelt es sich um ein laufend aktualisiertes Dokument. Der Inhalt wird im Verlaufe der Arbeit bei Bedarf angepasst. Beispielsweise wird der Risikoverlauf laufend aktualisiert. Die Aktualisierungen haben gemäss [1] zu erfolgen.

## 2.4. Referenzierte Dokumente

**Tabelle 4 Referenzierte Dokumente**

Referenz	Beschreibung
[1]	SCMP - Software Configuration Management Plan. Beschreibt die Art und Weise wie Dokumentänderungen gehandhabt werden.
[2]	Aufgabenstellung_DA06_V1.pdf, Aufgabenstellung gemäss 16.10.2006
[3]	Projektplanung-Java-Thread-Skalierung.mpp, Zeigt die Zeitliche Planung und den Verlauf des Projektes.

## 2.5. Definitionen und Abkürzungen

**Tabelle 5 Abkürzungen**

<b>Abkürzung</b>	<b>Beschreibung</b>
C	Controlling
CCM	Change & Configuration Management
DA	Diplomarbeit
HW	Hardware
SCMP	Software Configuration Management Plan
SPMP	Software Project Management Plan
STD	Software Test Document
SDD	Software Design Document
SW	Software

## 2.6. Links

**Tabelle 6 Links**

<b>Referenz</b>	<b>Beschreibung</b>
[ECLIPSE]	Freie Entwicklungsumgebung (hauptsächlich für Java aber über Erweiterungen auch für andere Sprachen gut geeignet): <a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
[VSTUDIO]	Microsoft Visual Studio 2005 Express Editions, frei verfügbare Entwicklungsumgebung: <a href="http://msdn.microsoft.com/vstudio/express/">http://msdn.microsoft.com/vstudio/express/</a>
[ANT]	Ant Java builder: <a href="http://ant.apache.org/">http://ant.apache.org/</a>
[LOG4]	Log4J, Log4cxx Applikations-Logging: <a href="http://logging.apache.org/">http://logging.apache.org/</a>
[CVS]	Concurrent Versioning System (CVS), <a href="http://www.cvshome.org/">http://www.cvshome.org/</a>
[TORTOISE]	TortoiseCVS, CVS GUI für Windows mit Explorer Shell-Integration: <a href="http://www.tortoisecvs.org/">http://www.tortoisecvs.org/</a>

## 3. Projektorganisation

### 3.1. Prozessmodell

Da die Aufgabenstellung in klar abgrenzbare Arbeitspakete mit offensichtlichen Abhängigkeiten gegliedert werden kann, wird das Projektresultat in einem sequenziellen Ablauf erarbeitet. Das Planungsmodell stützt sich somit auf das Wasserfall-Modell mit klar definierten Meilensteinen und Reviews, die Ergebnisse einzelner Projektphasen offenlegen. Punktuell können und sollen im Projektverlauf Iterationen möglich sein um die Qualität der Resultate zu verbessern (bspw. Entwicklung und Tests von Konzepten zur Parallelisierung).

### 3.2. Organisationsstruktur

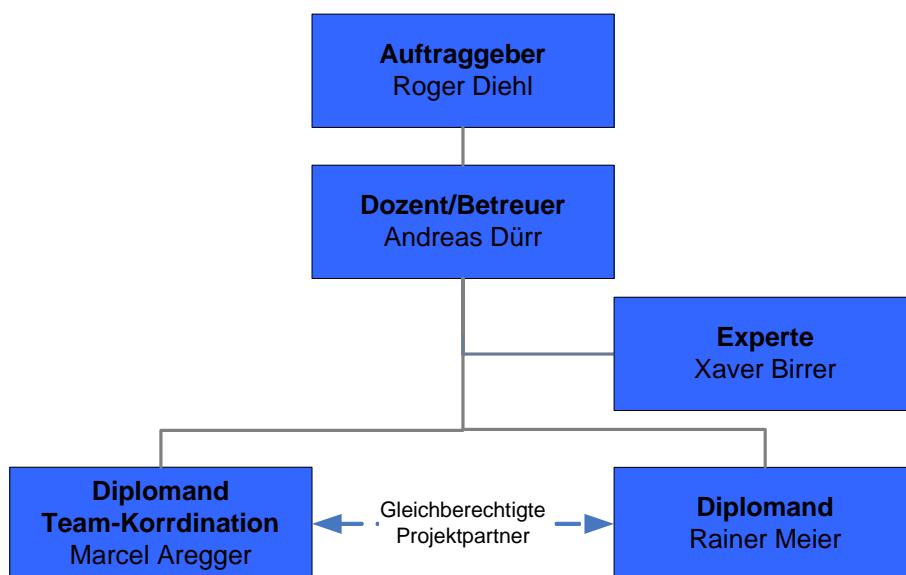


Abbildung 1 Projektorganisation

### 3.3. Organisatorische Abgrenzungen und Schnittstellen

#### Entscheidungs- und Eskalationsinstanz

Der Auftraggeber Herr Roger Diehl repräsentiert die Entscheidungs- und Eskalationsinstanz in diesem Projekt. Er entscheidet letztinstanzlich über Fragen in Bezug auf den Inhalt der Arbeit. Herr Andreas Dürr entscheidet in Fragen zur Methodik und zum Ablauf der Arbeit.

#### Verantwortungsbereiche

Die Projektergebnisse werden von den Diplomanden Marcel Aregger und Rainer Meier gemeinsam erarbeitet. Die Synchronisation geschieht hauptsächlich im direkten Kontakt. Dabei werden die in Kapitel 5.1 definierten Kommunikationsmittel eingesetzt.

### 3.4. Verantwortlichkeiten im Projekt

Die folgende Tabelle gibt eine grobe Übersicht über die Projektmitarbeiter und deren Verantwortlichkeitsbereiche:

**Tabelle 7 Projektverantwortlichkeiten**

<b>Name</b>	<b>Rolle</b>	<b>Verantwortlichkeitsbereich</b>	<b>Kompetenzen</b>
Roger Diehl	Auftraggeber	Auftraggeber und letztinstanzliche Entscheidungs- und Eskalationsinstanz.	Projektauftraggeber
Andreas Dürr	Dozent/ Betreuer	Direkte Eskalationsinstanz für die Projektmitarbeiter, Entscheidungsträger im Projektverlauf.	Direkte Entscheidungsbefugnis
Xaver Birrer	Experte	Fachliche Beurteilung der Projekt-ausführung	Direkte Entscheidungsbefugnis
Marcel Aregger	Diplomand	Planung und Realisierung des Pro-jektauftrages.	Problemeskalation
Rainer Meier	Diplomand	Planung und Realisierung des Pro-jektauftrages.	Problemeskalation

In der folgenden Tabelle werden sofern möglich eindeutig identifizierbare Rollen an Personen zugewiesen. Viele der zu erledigenden Arbeiten werden in gemeinsamer Arbeit erledigt und können nicht einzelnen Personen zugewiesen werden. Daher sind in der folgenden Tabelle nur Rollen aufgeführt, deren Verantwortung klar einer Person zugewiesen werden können.

**Tabelle 8 Rollenverteilung**

<b>Rolle</b>	<b>Name</b>	<b>Beschreibung</b>
CCM	Rainer Meier	Die Rolle des CCM (Change & Configuration Management) beinhaltet die Versionskontrolle und Unterhalt der dazu benötigten Werkzeuge sowie die Definition und Durchsetzung eines Versionierungskonzeptes. Dieses wird in [1] genauer beschrieben.
C	Marcel Aregger	Die Rolle des C (Controlling) beinhaltet die Überwachung, Dokumentation und Kommunikation des Projektfortschrittes.

Die Dokumentation wird generell gemeinsam erstellt. Gemäss den definierten Arbeitspaketen in der Projektplanung arbeitet jedes Projektmitglied seinen Anteil in das Gesamtdokument ein. Im Sinne der Konsistenz wird aber die Gesamtverantwortung über ein Dokument je einem Projektmitglied zugeordnet.

**Tabelle 9 Dokumentverantwortlichkeit**

Dokument	Verantwortung	Bemerkung
SCMP	Rainer Meier	MS Word
SPMP	Marcel Aregger	MS Word
Projektplan	Marcel Aregger	MS Project
Basisanalyse	Rainer Meier	MS Word
SDD	Rainer Meier	MS Word
STD	Marcel Aregger	MS Word
Conclusion	Marcel Aregger	MS Word

## 4. Management Prozess

### 4.1. Ziele und Prioritäten

Das Ziel der Arbeit ist die Analyse der Software-Skalierung auf Multi-Core/Multi-CPU Systemen gemäss der Aufgabenstellung [2].

In der Aufgabenstellung sind keine konkreten Prioritäten aufgeführt. Allerdings lässt sich aus der Aufzählung entnehmen, dass es bestimmte Themenbereiche optional sind.

**Tabelle 10 Zielpriorisierung**

Muss-Ziele	Kann-Ziele
Skalierung unter Windows XP (Betriebssystem-Ebene)	Skalierung unter UNIX (Betriebssystem-Ebene)
Skalierung von Java-Applikationen (Applikations-Ebene)	Skalierung von C++ Applikationen (Applikations-Ebene)

Muss-Ziele werden mit erster Priorität, Kann-Ziele mit zweiter Priorität behandelt. Konkret werden Prio-1 Ziele explizit in der Projektplanung berücksichtigt.

Der nachfolgende Zielkatalog ist gegliedert in Technische- und Methodische-Zielsetzungen und wurde aus der Aufgabenstellung abgeleitet.

#### 4.1.1. Technische Zielsetzungen

**Tabelle 11 Technische Zielsetzungen**

#	Zielsetzung	Kat.
{Z1}	Relevante Begriffe der Skalierung erläutern.	Muss
{Z2}	Relevante Grundlagen der Skalierung erläutern (bspw. Prozesse, Threads).	Muss
{Z3}	Bedingungen zur parallelen Abarbeitung einer Aufgabe aufzeigen.	Muss
{Z4}	Einsatzgebiete paralleler Anwendungen aufzeigen (bspw. Client, Server).	Muss
{Z5}	Software-Skalierung auf Java-Ebene (Basics, VM Klassen).	Muss
{Z6}	Software-Skalierung auf C++ Ebene (Basics, Libraries).	Kann
{Z7}	Skalierungstechniken auf Windows XP Ebene aufzeigen.	Muss
{Z8}	Skalierungstechniken auf Unix Ebene aufzeigen.	Kann
{Z9}	Skalierbare Prozessorarchitekturen aufzeigen (SMP).	Muss
{Z10}	Vergleich verschiedener Software-Plattformen.	Muss
{Z11}	Vergleich verschiedener Hardware-Plattformen.	Kann
{Z12}	Definition einer repräsentativen Testplattform (Hardware, OS, VM).	Muss
{Z13}	Testkonzept für verschiedene Designkonzepte definieren.	Muss
{Z14}	Suche und Evaluation von Test-Tools für skalierbare Anwendungen (Freeware).	Muss
{Z15}	Testklasse(n) zur Anwendung der verschiedenen Konzepte definieren.	Muss

{Z16}	Analyse der Konzepte anhand der Testklasse(n) auf Java-Basis (Implementierung).	Muss
{Z17}	Analyse der Konzepte anhand der Testklasse(n) auf C++-Basis (Implementierung).	Kann
{Z18}	Direkter Vergleich mehrerer Konzepte auf einer Plattform durchführen.	Muss
{Z19}	Direkter Vergleich der Plattformen mit einem ausgewählten Konzept durchführen.	Kann
{Z20}	Formulierung von Implementierungsempfehlungen (inkl. Code-Beispiele).	Muss

#### 4.1.2. Methodische Zielsetzungen

Tabelle 12 Methodische Zielsetzungen

#	Zielsetzung	Kat.
{Z21}	Inhalt der Teilprojektphasen durch Freigaben abgrenzen (Reviews, Meilensteine).	Muss
{Z22}	Projektdokumentation gemäss IEEE.	Muss
{Z23}	Projektplanung erstellen und permanente Fortschrittskontrolle.	Muss
{Z24}	Projektplanung durch Meilensteine strukturieren.	Muss
{Z25}	Implementierte Konzepte auf einer repräsentativen Plattform testen	Muss
{Z26}	Projektfortschritt mit Statusberichten dokumentieren.	Muss
{Z27}	Projektrisiken analysieren und dokumentieren.	Muss

#### 4.2. Annahmen und Abhängigkeiten

Um nach der Basisanalyse die praktischen Tests und Benchmarks durchführen zu können werden wir zu diesem Zeitpunkt entsprechende Multi-CPU/Multi-Core Systeme benötigen.

#### 4.3. Risikomanagement

Innerhalb des Risikomanagements sollen einige Risiken aufgezeigt werden, die das Projektresultat in qualitativer Hinsicht gefährden. Weiter sollen Aspekte aufgezeigt werden die den Verlauf des Projektes im Zusammenhang mit der Termintreue gefährden.

##### 4.3.1. Massnahmen

**Vermeidungsmassnahme:** Massnahmen, die zum Ziel haben, das Risiko zu senken, indem die Eintrittswahrscheinlichkeit gesenkt wird.

**Verminderungsmassnahme:** Massnahmen, die zum Ziel haben, das Risiko zu senken, indem das Schadenspotenzial gesenkt wird.

#### **4.3.2. Umsetzung und Wirksamkeit**

Die Umsetzung der Massnahmen sind in Bezug auf Komplexität und Aufwand unterschiedlich zu bewerten (leicht, mittel oder schwer). Die Wirksamkeit einer Massnahme wird gemessen am Potenzial die Eintrittswahrscheinlichkeit bzw. das Schadenspotenzial eines Risikos zu senken.

### 4.3.3. Identifizierte Risiken

Code	Risikobezeichnung	Verminderungsmaßnahme		Status der Maßnahmekette	Risikostatus
		(Massnahmen, die zum Ziel haben, das Risiko zu senken, indem das Schadenspotenzial gesenkt wird)	Status Umsetzung		
	Risiken Projektresultat	Entwickler-Potenzial schenkt mehr Zeit für die Erfüllungswahrscheinlichkeit (Massnahmen, die zum Ziel haben, das Risiko zu senken, indem das Schadenspotenzial gesenkt wird)	Geachtet Vorförk Sammel	Statis Us Umsetzung	1,2, 3 0,2-1,0
1 T / I	Testresultate können nicht interpretiert werden ohne tiefere HW- und OS-Analyse	50 80 Schwerpunkt auf HW- und OS-Analyse legen. Analysetools mit erweiterter Funktionalität evaluieren	3 0,5 Reproduzierbarkeit sicherstellen durch Beschreibung Testparameter (Nachfolgetest)	1 0,8 76,7	20,7
2 T / E	Durch mangelnde Verfügbarkeit der Hardware- plattform (HTA) kann Testserie nicht vnlständig durchgeführt werden	40 50 Frühzeitige Definition und Beschaffung der Testplattform (OS und Hardware)	1 0,8 Alternative Beschaffungsoptionen festlegen (Eigenlieferung)	2 0,8 80,0	17,0
3 Q / I	Projektresultat-/Output entspricht nicht den Vorstellungen der Auftraggeber (Fehlinterpretation der Aufgabenstellung)	20 60 Erstellen eines Zelkkataloges mit klassifizierten Zielsetzungen. Freigabe der Zielsetzungen in der Startphase des Projektes	1 0,9 Sollte Grundlagenanalyse die abgesehen vom Projektergebnis einen Nutzen darstellt (Unterricht HTA)	2 0,7 76,7	15,7
4 T / E	Freeware Test-Tools entsprechen nicht den Anforderungen gemäss Festkonzept	50 40 Testgrößen und -verfahren definieren die den Einsatz von optionalen Tools und Techniken zulässt	2 0,6 Reproduzierbarkeit sicherstellen durch Beschreibung Testparameter (Nachfolgetest)	1 0,8 66,7	15,7
5 Q / I	Nachvollziehbarkeit der Skalierung ist aufgrund der Komplexität oder Methode nicht gegeben	50 50 Skalierbarkeit visuell unterstützen	1 0,7 Dokumentation der eingesetzten Testtools. Transparente Dokumentation der Testreihe	1 0,6 43,3	14,3
	Risiken Projektverlauf				
6 Q / I	Phasengerechte Dokumentation kann infolge Dokumentationsaufwand nicht eingehalten werden	30 60 Dokumentationsaufwand in Projektplanung berücksichtigen. Erstmalige Dokumentation mit Tools/Methoden sicherstellen (bspw. CvS)	1 0,7 Permanenter Dokumentenfluss zum Auftraggeber sicherstellen (Zu verfügbarenstellen von Zwischenresultaten)	1 0,7 46,7	13,7
7 Q / I	Kommunikationsprobleme im Projektumfeld schwächen die Teamleistung	30 40 Häufige und direkte Kommunikation mit Auftraggeber (geplante Reviews, Statusberichte, etc.)	1 0,7 Umfassende und transparente Projektdokumentation	2 0,5 56,7	12,7
I	Internes Risiko	O Organisationsrisiko	1 Umsetzung Leicht		
E	Externes Risiko	Q Qualitatives Risiko	2 Umsetzung Mittel		
T	Technisches Risiko		3 Umsetzung Schwier		

#### 4.3.4. Risikolandschaft

Die Risikoverteilung zeigt die relative Lage identifizierter Risiken im Projekt. Aus dem Portfolio kann entnommen werden, dass nicht alle Risiken gleichermassen kritisch sind. In Bezug auf die Dringlichkeit von (Gegen)Massnahmen ergeben sich 3 signifikante Gruppen:

- Quadrant 1: Sofortige Umsetzung der Vermeidungsmassnahmen, permanente Kontrolle
- Quadrant 2,3: Umsetzung der Vermeidungsmassnahmen, punktuell Kontrolle
- Quadrant 4: Umsetzung der Vermeidungsmassnahmen, phasenweise Kontrolle

Es wurden 7 Hauptrisiken identifiziert, wovon 3 Risiken im Quadranten 1 und 2 liegen.

Das Risiko „Interpretation der Testresultate“ (1) gilt hier als „hoch risikoreich“ und beschreibt die Gefahr, dass Resultate in der Testserie aufgrund der Komplexität (HW oder OS) nicht richtig interpretiert werden können.

„Unbefriedigender Projektorput“ (3) und „Phasengerechte Dokumentation“ (6) sind ebenfalls Risiken, deren Verlauf bzw. Gegen-Massnahmen erhöhte Aufmerksamkeit geschenkt werden muss.

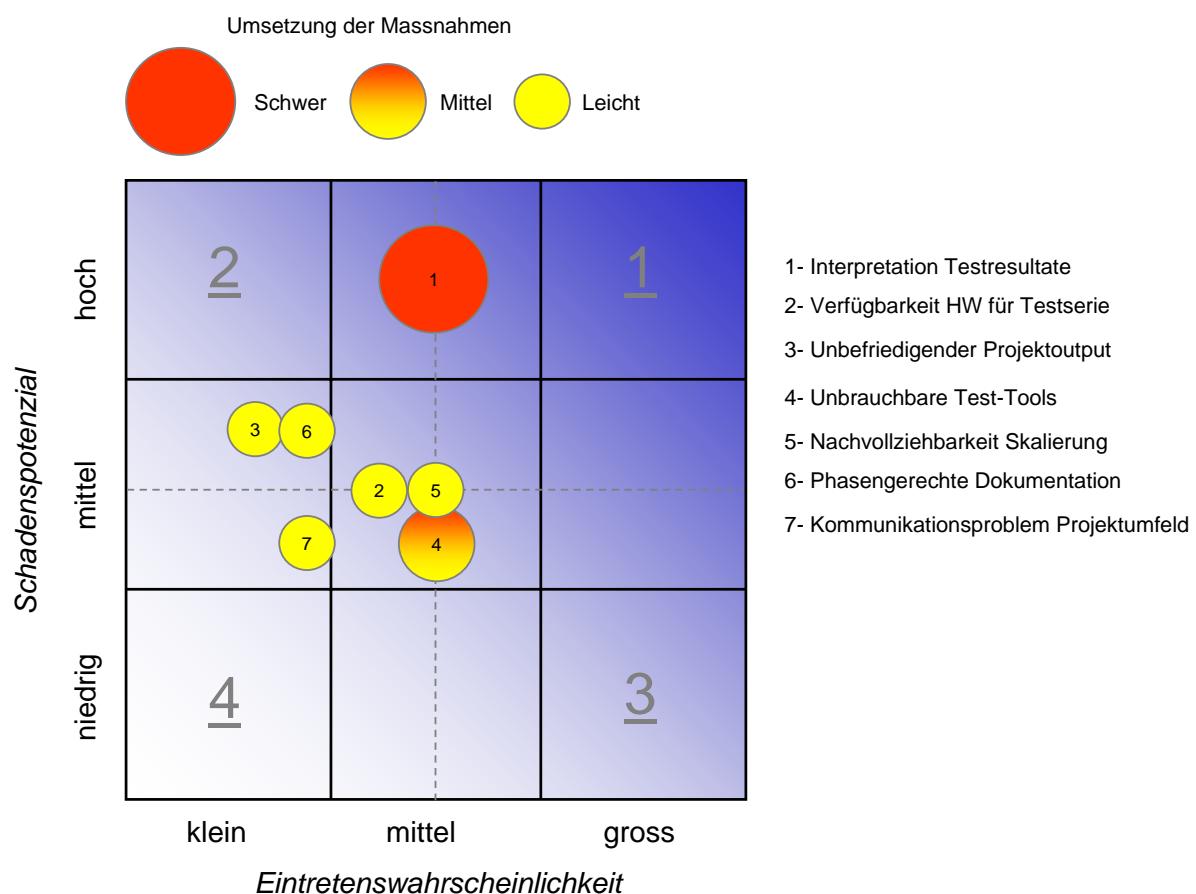


Abbildung 2 Risikoportfolio

#### **4.4. Monitoring und Controlling**

Die Diplomanden arbeiten weitgehend selbstständig. Weitreichende Massnahmen die für ein effizientes Controlling notwendig wären, sind unter Berücksichtigung der kurzen Projektdauer nicht gerechtfertigt. Der primäre Fokus liegt daher auf:

- Den festgelegten Meilensteinen (Termin)
- Den festgelegten Reviews (Inhalt)
- Der Kommunikation des Projektstatus (Reporting)

Die Daten der Reviews und Meilensteine sind unter [3] dokumentiert..

## 5. Technischer Prozess

### 5.1. Methoden, Tools, Techniken

Gemäss Projektauftrag [2] werden keine Vorschriften in Bezug auf Methoden, Tools und Techniken gemacht. Daher führen wir hier selber einige bevorzugte Tools auf, diese werden wo sinnvoll und nötig eingesetzt.

#### Entwicklungsumgebungen

---

Eclipse (für Java und eventuell auch für C/C++), [ECLIPSE]

Microsoft Visual Studio Express Edition (für C++ Entwicklung, optional): [VSTUDIO]

#### Unterstützende Werkzeuge

---

Ant, [ANT]

Log4J/Log4cxx, [LOG4]

#### Debugger

---

Eclipse Debugger

Visual Studio Debugger

#### Performance und Low-Level Messungen:

---

[noch unbekannt]

#### Source-Code-Verwaltung

---

CVS, [CVS]

TortoiseCVS, [TORTOISE]

## **Kommunikation**

---

Direkte Zusammenarbeit direkt an der HTA

Telefon, E-Mail,

Skype

Instant-Messenger

## **Dokumentverwaltung**

---

Dokumente werden in CVS verwaltet. TortoiseCVS ([TORTOISE]) wird als Windows-GUI verwendet.

### **5.2. Software Dokumentation**

Software wird soweit möglich direkt im Sourcecode dokumentiert. In Java geschieht dies im Code mit JavaDoc Kommentaren. Codeschnipsel werden in den Dokumenten genauer erklärt. Zur Erklärung der Abläufe werden sofern nötig UML-Diagramme erstellt.

### **5.3. Projektunterstützung**

Zur Beschaffung der benötigten Test-Hardware mit Multi-CPU bzw. Multi-Core Ausrüstung kann mit Bruno Joho Kontakt aufgenommen werden.

## 6. Arbeitspakete und Zeitplan

### 6.1. Arbeitspakete

Angelehnt an die unter Punkt 2.1.3 definierten Schwerpunkte können folgende Arbeitspakete definiert werden:

**Tabelle 13 Arbeitspakete**

Arbeitspaket	Beschreibung	Aufwand [h]
Projektmanagement	Planung, Controlling Projektverlauf	16
Basisanalyse	Erarbeitung Grundlagen HW und SW	120
Evaluation	Testplattform, Testkonzept und Tooling	46
Umsetzung/Implementierung	Testplattform, Implementierung Konzepte, Testing	88
Auswertung	Interpretation der Resultate, Empfehlungen	68
Dokumentation	Dokumentation der DA	60
Projektabchluss	Webabstrakt, Diplomposter, Präsentation	22

Die nachfolgende Auflistung zeigt deutlich die Schwerpunkte der Aufgabenstellung aus Sicht Projektteam. Die Basisanalyse also die Erarbeitung der notwendigen Grundlagen nimmt mit rund 28% einen sehr grossen Stellenwert ein. Umsetzung/Implementierung und Auswertung stellen im Verbund mit rund 37% ebenfalls ein „Schwergewicht“ dar.

- Basisanalyse 120 h (~28%)
- Evaluation 46 h (~11%)
- Umsetzung/Implementierung 88 h (~21%)
- Auswertung 68 h (~16%)
- Dokumentation 60 h (~14%)

### 6.2. Abhängigkeiten

Die Arbeitspakete bzw. deren Resultate sind in der Regel voneinander abhängig und müssen in der richtigen Reihenfolge abgearbeitet werden. Einige Unter-Aufgaben lassen sich unabhängig voneinander betrachten und können somit parallel abgearbeitet werden. Beispielsweise können mehrere Personen gleichzeitig an unterschiedlichen Testreihen arbeiten. Auch der Vergleich der Plattformen basierend auf den Testreihen kann parallel zum Vergleich der Konzepte geschehen.

### 6.3. Benötigte Ressourcen

Für die Umsetzungs- bzw. Implementierungsphase werden Test-Plattformen benötigt um die Tests gemäss dem Testkonzept durchführen zu können (Hardware, Betriebssystem, Tools).

## 6.4. Zeitplan

### Arbeitspakete

Die Ablaufplanung (Extrakt aus Project-Planung) zeigt die Haupt-Aufgabenpakete über den gesamten Projektverlauf. 3 Meilensteine grenzen die Phasen „Basisanalyse“, „Umsetzung/Implementierung“ und „Auswertung“ ab.

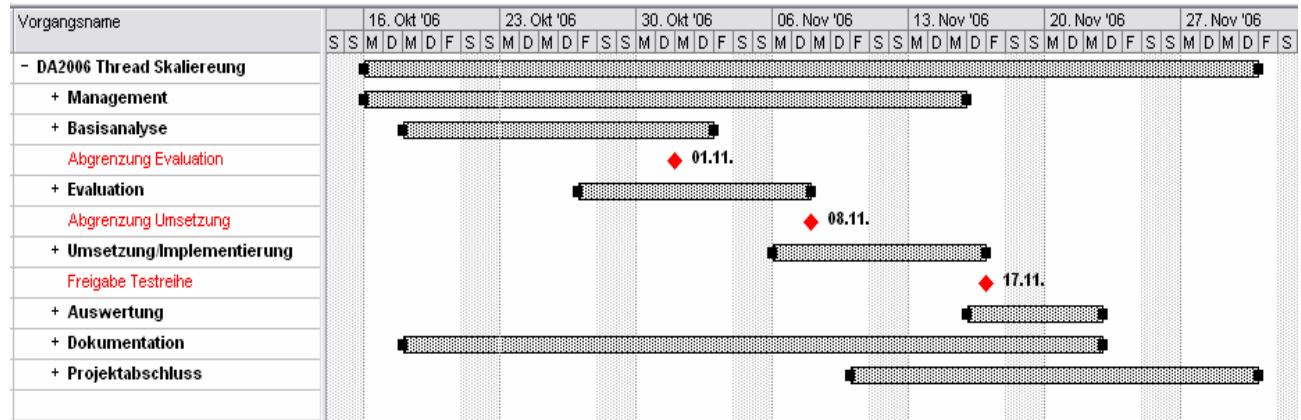


Abbildung 3 Ablaufplan

### Reviews

Die weitere Detaillierung zeigt die 4 geplanten Reviews über den gesamten Projektverlauf.

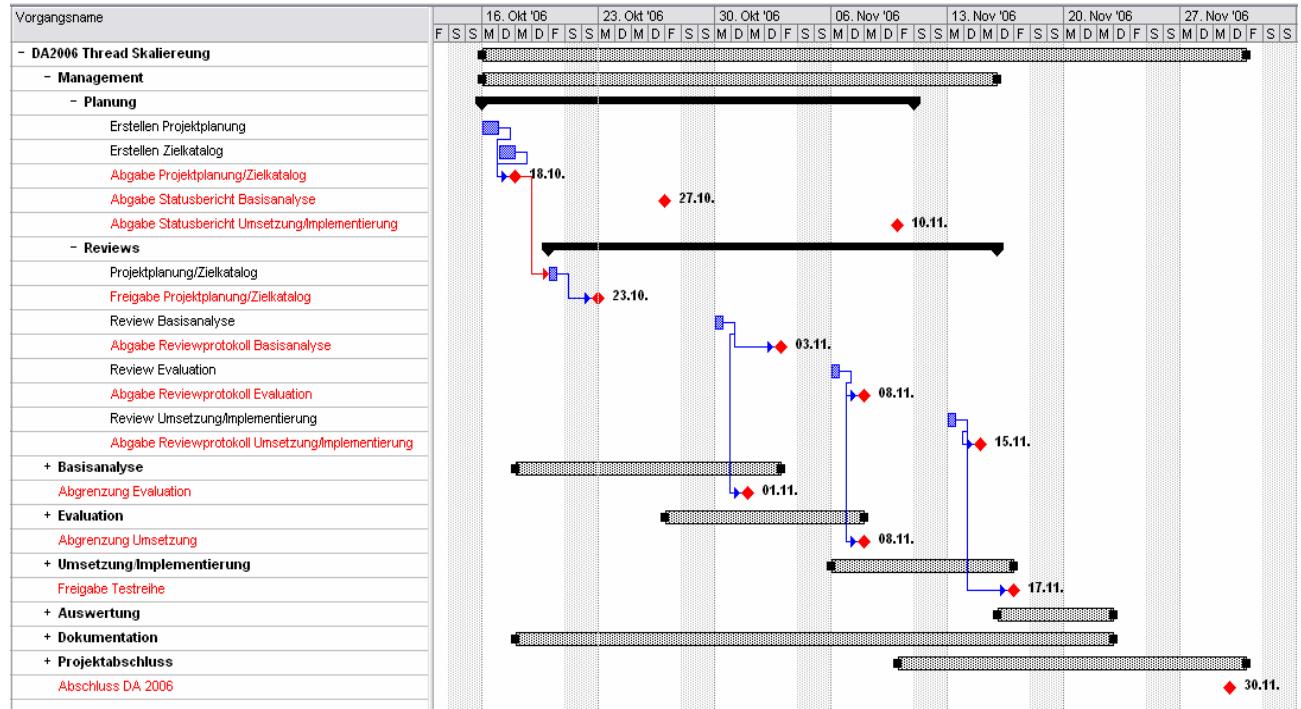


Abbildung 4 Ablaufplan detailliert

### 6.4.1. Reviews

Nr:	Datum	Beschreibung
1	20.10.2006	Projektplanung und Zielkatalog
2	30.10.2006	Review Basisanalyse
3	06.11.2006	Review Evaluation
4	13.11.2006	Review Umsetzung und Implementierung

### 6.4.2. Meilensteine

Folgende Meilensteine sind für diese Projektarbeit vorgesehen:

#### Projektverlauf

Nr:	Datum	Beschreibung
1	18.10.2006	Abgabe Projektplanung/Zielkatalog
2	23.10.2006	Freigabe Projektplanung/Zielkatalog
3	01.11.2006	Abgrenzung Evaluation
4	08.11.2006	Abgrenzung Umsetzung
5	17.11.2006	Freigabe Testreihe

#### Reporting

Nr:	Datum	Beschreibung
6	27.10.2006	Abgabe Statusbericht Basisanalyse
7	10.11.2006	Abgabe Statusbericht Umsetzung/Implementierung

#### Projektabchluss

Nr:	Datum	Beschreibung
8	13.11.2006	Vorab-Präsentation Diplomarbeit
9	24.11.2006	Abgabe Webabstrakt
10	20.11.2006	Abgabe Diplomposter
11	30.11.2006	Aufhängen Diplomposter
12	30.11.2006	Abschluss DA 2006

Die Reviewprotokolle werden jeweils vom betreuenden Dozenten erstellt und verteilt.

### 6.4.3. Dokumente

Die Dokumentabgabe ist wie folgt geplant:

**Tabelle 14 Dokumente, Abgabetermine**

Dokument	Abgabedatum Draft	Abgabedatum Final
SCMP	18.10.2006	25.10.2006
SPMP	18.10.2006	25.10.2006
Projektplan	18.10.2006	25.10.2006
Basisanalyse	27.10.2006	03.11.2006
SDD	06.11.2006 10.11.2006	24.11.2006
STD	10.11.2006	24.11.2006
Conclusion		24.11.2006

## 7. Verzeichnisse

### 7.1. Tabellenverzeichnis

Tabelle 1 Projektschwerpunkte .....	5
Tabelle 2 Projektergebnisse .....	6
Tabelle 3 Projektergebnisse 2 .....	6
Tabelle 4 Referenzierte Dokumente .....	6
Tabelle 5 Abkürzungen .....	7
Tabelle 6 Links .....	7
Tabelle 7 Projektverantwortlichkeiten .....	9
Tabelle 8 Rollenverteilung .....	9
Tabelle 9 Dokumentverantwortlichkeit .....	10
Tabelle 10 Zielpriorisierung .....	11
Tabelle 11 Technische Zielsetzungen .....	11
Tabelle 12 Methodische Zielsetzungen .....	12
Tabelle 13 Arbeitspakete .....	19
Tabelle 14 Dokumente, Abgabetermine .....	22

### 7.2. Abbildungsverzeichnis

Abbildung 1 Projektorganisation .....	8
Abbildung 2 Risikoportfolio .....	15
Abbildung 3 Ablaufplan .....	20
Abbildung 4 Ablaufplan detailliert .....	20

### 7.3. Index

Abhängigkeiten .....	12, 19	Eclipse .....	7, 17	Projektübersicht .....	5
Abkürzungen .....	7	Entwicklungsumgebung .....	7	Projektunterstützung .....	18
Aktualisierungswesen .....	6	Links .....	7	Prozessmodell .....	8
Annahmen .....	12	Log4J .....	7	Referenzen .....	6
Ant .....	7, 17	Methoden .....	17	Ressourcen .....	19
Arbeitspakete .....	19	Monitoring .....	16	Reviews .....	16
Aufgabenstellung .....	6	Organisationsstruktur .....	8	Risikomanagement .....	12
Basisanalyse .....	22	Prioritäten .....	11	Rollenverteilung .....	9
Conclusion .....	22	Projektergebnisse .....	6	Schnittstellen .....	8
Controlling .....	16	Projektorganisation .....	8	SCMP .....	6, 22
CVS .....	7, 17	Projektplan .....	22	SDD .....	22
Definitionen .....	7	Projektplanung .....	6	Skype .....	18

SPMP.....	22	TortoiseCVS.....	7, 17	Visual Studio.....	7, 17
STD.....	22	UML.....	18	Zeitplan.....	19
Techniken.....	17	Verantwortlichkeiten .....	8	Ziele .....	11
Tools .....	17	Verantwortung .....	9	Zielkatalog.....	11