Software Project Management Plan

Eindhoven, December 1, 2009 SPMP-3.0.1869

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Project Manager:
Wilco Belgraver Thissen, 0514143

Quality Assurance Manager:
Jelle Hellings, 0592127

Senior management:
Mark van den Brand, HG 5.59
Lou Somers, HG 5.36

Advisor:
Erik Luit, HG 7.12

Customer:
Natalia Sidorova, HG 7.84

Project team:
Roy Berkeveld, 0608170
Gijs Direks, 0611093
Michael van Duijkeren, 0535368
Neal van den Eertwegh, 0610024
Dion Jansen, 0590077
Koen Kivits, 0608715
Sander Leemans, 0608896
Kevin van der Pol, 0620300
Nick van der Veeken, 0587266

Computer Science, TU/e

Abstract

This is the Software Project Management Plan (SPMP) for the "GROUP QIS" project group, complying with the Software Engineering Standard as set by the European Space Agency [1]. In this document, the estimated time and cost, and the planning of the entire project are provided. In addition, an overview is given of possible risks that can endanger the delivery of the product within time and as agreed. The SPMP is used by the Project Manager (PM) to guide the project and to come to an agreement with Senior Management about budgets and planning. The PM uses the SPMP to organize the project in different phases and to create deadlines. The SPMP is closely related to the Software Quality Assurance Plan (SQAP).

Contents

Document Status Sheet

Document status overview

General

Document title:	Software Project Management Plan
Identification:	SPMP-3.0.1869
Author:	Wilco Belgraver Thissen
Document status:	Final

Document history

Version	*Date*	*Author*	*Reason of change*
0.1	16-09-2009	Wilco Belgraver Thissen	First review version
0.2	4-10-2009	Wilco Belgraver Thissen	Second review version
1.0	4-10-2009	Wilco Belgraver Thissen	Internally accepted
1.1	6-10-2009	Wilco Belgraver Thissen	Extra unavailabilities added
2.0	6-10-2009	Wilco Belgraver Thissen	Again Internally accepted
3.0	24-11-2009	Wilco Belgraver Thissen Several project members	Internally accepted after addition of Implementation process chapter

Document Change Records since previous issue

General

Datum:	2009-12-01
Document title:	Software Project Management Plan
Identification:	SPMP-3.0.1869

Changes

Page	*Paragraph*	*Reason to change*
many	*many*	Internal review
Unavailabilities	*Table 5.5.3*	Extra unavailabilities added
Appendices	*added for upcoming period*	Nearing of periods
Implementation process	*Chapter 6*	Added chapter on implementation process, details how and what has to be done.
3.4 *5.5.1*	*Section for weekly delivery added* *ADD deadline added*	Internal review

Chapter 1

Introduction

1.1 Project overview

The GROUP QIS has been formed at the Eindhoven University of Technology (TU/e) as part of the Software Engineering Project of fall 2009. The goal of the group is to create a management system which will allow several individuals, such as the faculty director and the people in charge of the timetable creation to easily do their part of the scheduling. The new system will replace the current one and as such it should contain the scheduling information which is currently available in addition to the required functionality.

The objective is to create this application before January 15th, 2010. In addition several other deliverables are due on earlier dates. All the work should be done within the allotted time, which is 3304 hours.

The main deliverables are the User Requirements Document, the Software Requirements Document, the Architectural Design Document, the Detailed Design Document and the software source code itself. The project is split up in several phases, each evolving around the production of one of these documents. Additionally, several management documents will need to be written, maintained and adhered to throughout the project.

1.2 Project deliverables

During the course of the project, several documents have to be produced and delivered to the client and to senior management. All these documents have to be written according to the ESA software engineering standards [1]. All products that will be delivered to the client are mentioned in the table 1.1, together with the phases for which they represent outputs.

The client intends to develop the software further after this project. Therefore the client will receive a copy of all project documents, except for the management documents. All documents have to be delivered both on paper as well as in electronic form.

Phase	Deliverables	To whom
User Requirements	Software Project Management Plan (SPMP)	Senior Management
	Software Quality Assurance Plan (SQAP)	
	Software Verification and Validation Plan (SVVP)	
	Software Configuration Management Plan (SCMP)	
	User Requirements Document (URD)	Senior Management and Client
	Acceptance Test Plan (ATP)	
Software Requirements	Software Requirements Document (SRD)	Senior Management and Client
	System Test Plan (STP)	
Architectural Design	Architectural Design Document (ADD)	Senior Management and Client
	Integration Test Plan (ITP)	
Detailed Design and Production	Detailed Design Document (DDD)	Senior Management and Client
	Unit Test Plan (UTP)	
	Software User Manual (SUM) Software Transfer Document (STD)	
	Source Code	

Table 1.1: Deliverables per phase and recipient

1.3 Evolution of the SPMP

This document is subject to changes. The assumptions, dependencies and constraints for the project, the detailed time and resource planning for each phase (see appendices) can change during the project. Changes in this document will lead to a SPMP with a new version number, but with the same status. However, if these changes lead to changes in the milestones planning of the project, described in section 5.5, these changes have to be discussed first with the Senior Management, before they can be incorporated in the document. This will be done during progress meetings. The detailed planning for each phase is described in the appendices of this document. These appendices are updated at different moments in time during the project, but before the start of the phase they refer to.

1.4 List of definitions

2IP35	The Software Engineering Course
AD	Architectural Design phase
ADD	Architectural Design Document
AT	Acceptance Test
ATP	Acceptance Test Plan
Client	Natalia Sidorova
CM	Configuration Manager
DD	Detailed Design phase
DDD	Detailed Design Document
ESA	European Space Agency
ECP	External Contacts Person
TU/e	Eindhoven University of Technology
PM	Project Manager
QM	Quality Manager
SCMP	Software Configuration Management Plan
SEP	Software Engineering Project
SM	Senior Management
SPMP	Software Project Management Plan
SQAP	Software Quality Assurance Plan
SR	Software Requirements definition phase
SRD	Software Requirements Document
STD	Software Transfer Document
SUM	Software User Manual
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
TR	Transfer phase
UR	User Requirements definition phase
URD	User Requirements Document
VQM	Vice Configuration Manager
VPM	Vice Project Manager
VQM	Vice Quality Manager

1.5 List of references

[1] ESA Board for Software Standardization and Control (BSSC). European space agency software engineering standards, February 1991. (ESA PSS-05-0 Issue 2).

[2] Group QIS. Software configuration manangement plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

[3] Group QIS. Software quality assurance plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

[4] Group QIS. Software verification and validation plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

[5] Group QIS. User requirements document. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

Chapter 2

Project organization

2.1 Process model

The project will mostly follow the waterfall model. In this model, every next phase can only begin when all preceding phases have been successfully completed. However, when necessary, we will let some of these phases run in parallel to decrease the delay due to minor problems with the preceding phases. The phases of this project are:

- User Requirements (UR)
- Software Requirements (SR)
- Architectural Design (AD)
- Detailed Design (DD)
- Transfer (TR)

The UR phase involves creating the management documents (SPMP, SCMP, SVVP and SQAP) and the URD. Further, the CM team has to make sure all necessary hardware and software is available.

The DD phase involves creating the source code and performing the unit, integration and system tests. This phase we will deviate from the waterfall model to do the coding incrementally. Because of this it will start early to be done partially parallel to the AD phase.
The incremental coding means that a small product is created, which is stable but doesn't do very much. Then, a few new features are added and the next stable version is released. This works especially well with a priority system, the highest priority features are implemented in the earliest versions.
Acceptance tests are performed during the TR phase.

Although the ESA Software Engineering Standard prescribes a sixth phase (the maintenance phase), we allow ourselves to omit this phase. This is done because the project will terminate after the TR phase is completed.

2.2 Organisational structure

The TU/e employs the Senior Management and the technical advisor. Communication between Senior Management and the project group is done via the PM. Otherwise, only the VPM can contact the Senior Management directly and only when he is concerned about the functioning of the PM. Whenever the PM is not able to perform his duties for a certain period of time, this is done by the VPM.

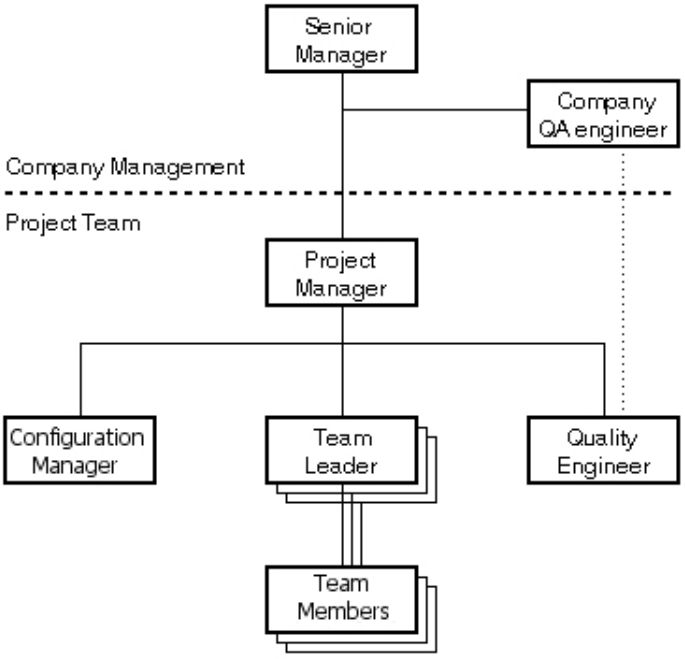

Figure 2.1: organisational structure

Figure 2.1 shows the organizational structure of the project. The PM, QM and CM will be supported by the VPM, VQM and VCM respectively.

2.3 Boundaries and interfaces

There are several persons and groups for the project group to interact with. These are:

- The university: The Software Engineering Project (SEP) is a project of TU/e. The PM and the QM are responsible for, respectively, the project progress and its quality, and report to the SM. The university is the owner of the software and documents produced in this project.
- The technical advisor: is a staff member of the Computer Science department of TU/e. Team members may consult him on technical issues. The technical advisor is generally present at the weekly meeting.
- The customer: Natalia Sidorova, who requested the system. The customer will always be contacted by the ECP.

Due to the small scale and duration of the project, no direct relationship with end users and subcontractors can be defined.

2.4 Project responsibilities

In the project, certain management tasks need to be performed. The risk exists, that a member of the team with a management function, is suddenly unavailable. Therefore every manager will be assisted, and if necessary, replaced by a vice-manager.

Project Manager (PM)

The PM is Wilco Belgraver Thissen, Koen Kivits is assisting as VPM. The primary goal of the project manager is to ensure the completion of the project in the available time, within budget. The PM makes the planning and ensures that it is feasible. Whenever problems occur the planning must be adapted in order to obtain a feasible planning again. The PM must ensure the progress of the project. He will be assisted by the vice PM. In case of absence of the PM, the vice PM takes over the PM's tasks. The tasks the PM must perform are:

- Writing this SPMP
- Being the chairman during meetings
- Motivating team members
- Forming teams and assigning tasks
- Checking progress
- Managing the time budget
- Defining work packages and goals
- Providing feedback to the Senior Management through progress reports

Quality Manager (QM)

The QM is Jelle Hellings, Nick van der Veeken is assisting as VQM. The primary goal of the quality manager is to ensure the quality of the end product and the overall process. In case of absence of the QM, the vice QM takes over the QM's tasks. The tasks the QM must perform are:

- Writing the SQAP [3]
- Verifying that procedures and standards which are defined in the SQAP are adhered to
- Checking that all project documents are consistent
- Arranging internal and external reviews
- Monitoring and reviewing all testing activities

- The vice QM is responsible for making the minutes and assuring they are available as soon as possible

Configuration Manager (CM)

The CM is Gijs Direks, Roy Berkeveld is assisting as VCM. The task of the Configuration Manager is to create and maintain the project library. More information about the project library can be found in SCMP[2]. In case of absence of the CM, the vice CM takes over the CM's tasks. The tasks the QM must perform are:

- Writing the SCMP
- Creating a repository for all documents and code
- Checking that the repository is used appropriate (that is according to the SCMP) by all team members
- Maintaining the repository according to the SCMP

Team Leader

During the project, several teams will exist. A team leader is responsible for the activities of his team. The tasks a team leader must perform are:

- Planning and coordinating team activities
- Providing feedback about team progress to the PM
- Motivating team members
- Chairing internal reviews of the items made by his team

Team Member

The team members are all members of a team including the team leader. The tasks a team member must perform are:

- Assisting the Team Leader or Project Manager by signaling problems in an early stage
- Executing plans made by the Team Leader and by the Project Manager
- Keeping track of time spent on various tasks
- Following procedures and plans

Chapter 3

Managerial process

3.1 Objectives and priorities

The management objective is to deliver the product in time and of high quality. The PM and QM work together to achieve this by respectively checking that progress is made as planned and monitoring the quality of the product at various stages.

3.2 Assumptions, dependencies and constraints

Partly due to the educational nature of the project, it is limited by a rather large number of factors:

- The budget is (9 team members * 12 ECTS * 28 hours) = 3024 working hours
- The PM and QM budget of (5 ECTS * 28 hours) = 140 hours. About 75% of this time should be spend on the actual project
- The following hard deadlines, set by senior management:
  - Intermediate presentation: November 9th, 2009
  - Final presentation: January 11th, 2010
  - Project deadline: January 15th, 2010
  - Documentation deadline URD: October 4th 2009
  - Documentation deadline SRD November 1st 2009
- Holidays weeks: December 21th - January 3th. There are also two non-educational Fridays at October 16th and December 11th
- Exam periods: October 26 - November 7th and January 18th - January 30th
- Scheduled lectures and homework for classes other than this project - different for every team member

- Planned holidays and educational travels of the team members and customer. See section 5.5.3

3.3 Risk management

Miscommunication
Probability: High
Impact: High
Misinterpretations of what other team members say and write might stand in the way of a common understanding of what to do and how to do it. This might lead to delays, unwanted results and double work.
Prevention: Throughout the project, and especially during weekly meetings, the PM has to make sure every team member understands the task given to him, by having frequent talks with each group member about their task. It is important that the minutes of the meeting are correct and complete, and they should be read by everyone with care.
Correction: When a problem occurs, the PM arranges a meeting with all involved people to come to a common understanding of the situation. After this meeting, its results are briefed to all team members.

Too many planned features lead to infeasible design
Probability: Medium
Impact: Medium
Prevention: The technical advisor should be consulted on whether the delivery of the planned product can be done within the time budget. Every item should have a priority.
Correction: By closely monitoring progress the decision to drop certain requirements can be made in time.
Note: Due to the incremental coding approach, this is less of a problem than it otherwise might have been.

Illness or absence of team members or the PM
Probability: Medium
Impact: High
Prevention: Team members should warn their team leader or the PM timely before a planned period of absence. The PM should make planned absence a point on the first meeting to make sure that absence that is known at that time is taken into account in the schedule.
Correction: Every role has a person to replace him. Communication between these two people is very important. The "vice" person should be actively involved in all work in order to be able to replace his counterpart. All important information and design decision should be documented in documents or minutes, to make sure that as little information as possible is lost.

Loss of critical information, documents or code
Probability: Medium
Impact: High
Prevention: The SCMP[2] should be written and used with care. The base assumption should be

that there is a backup of every single piece of information at any single time.
Correction: Use the latest backups to recover the most recent version. If the missing parts are necessary, replace these as soon as possible.

The customer changes his mind about the requirements, or there is disagreement about the requirements interpretation.
Probability: High
Impact: Medium
Prevention: It should be made very clear to the customer that after a certain date the requirements can't change anymore
Correction: If the customer changes his mind during the UR phase his new requirements can be incorporated in the URD [5]. Procedures in SQAP [3] detail if the URD may be changed after approval, and (if so) how to implement changes.

The customer is not available when needed
Probability: Medium
Impact: Medium
Prevention: Meetings with the customer should be planned well in advance. The customer has been given room in his schedule for his Software Engineering related work. Holidays and other travel plans of all people involved should be put in this document in section 5.5.3.
Correction: When the customer is not available, meetings may have to be rescheduled.

Imbalance in work load
Probability: Low
Impact: Medium
Prevention: Keep track of the work done by individuals by using logs.
Correction: Redistribute the work when large differences appear.

Time shortage
Probability: High
Impact: Medium
Prevention: Care is taken to plan enough spare time. For every external review two dates are planned in case the first review fails to lead to acceptance. Planning is based on experience from previous projects.
Correction: When tasks fail to be finished in time or when they are finished earlier than planned the project planning is adjusted. If time shortage occurs, several user requirements, which have lower priority, can be dropped after consultation with the client. Only those of highest priority are mandatory. Implementation of the lowest priority requirements is not expected to be reached.

Design Errors
Probability: Medium
Impact: High
Prevention: The design should be reviewed very critically. An advisor should be consulted frequently on his opinion about the feasibility and the correctness of certain design decisions. The

external reviews of the systems design will also ensure that the team will not commence with an erroneous design.
Correction: When errors in the design are noticed an advisor should be consulted to help correct the design errors as soon as possible. Also all work, which depends on the faulty design, should be halted until the error is corrected.

Personal conflicts between team members
Probability: Low
Impact: Medium
Prevention: By stimulating the communication between team members, conflicts may be avoided. Also team-building activities should be held.
Correction: Team members who are known to have had conflicts between them will not be assigned to the same team. Furthermore, the PM will discuss the situation with the team members in an effort to clear the air.

Lack of expertise to fulfill certain tasks
Probability: High
Impact: Low
Prevention: By focusing on solutions for which the necessary expertise is available and sharing expertise available within the team this risk may be reduced.
Correction: In SQAP, methods for acquiring skills are detailed.

3.4 Monitoring and controlling mechanisms

The monitoring of progress is done by the PM using the following means:

Weekly Project Group Meetings The project group meetings take place in the project room, HG10.56. Project group meetings usually take place on Tuesday at 12.45, although this time may be subject to change. Emails will be send about the time if it changes. These meetings are meant to inform each other of the progress made on various tasks. New tasks are assigned by the PM on these meetings. Before the meeting, all members read minutes of previous meeting. The PM takes care of the agenda and presides the meeting.

Note that much of the content of these meetings can also be discussed informally. These meetings are primarily used to bring everybody up to speed with the current status.

Weekly increment delivery Week 10 of the project the first increment is started. From this moment till end of coding every Monday at 13:00 a group member together with PM and QM visits the SM to show the progress of the past week. On the weeks that also the progress meeting is it is after this meeting with the same group member. A couple of items have to be delivered at this meeting:

- List of functionality realised this week that can be checked as well as the lists of previous weeks. This list includes referenced to the implemented requirement.
- A printed version of the URD
- Url of the increment to review and of the 2 previous increments.

At this meeting the SM checks the new items and a few random checks of existing functionality to confirm nothing has been broken.

Progress Meetings These meetings are scheduled biweekly at Monday 13:00 in room 5.36. On these meetings the PM, a team member and if available QM meet with the SM. Before progress meetings the following things need to be done:

- Write a progress report after the example of the previous reports.
- Read the minutes of the previous meeting.
- Deliver the report to the SM half an hour before the start of the first meeting on that day.

Project metrics Every week, the work done by the members, needs to be administrated. Each team member has to fill in their hours on a webbased log. This log needs to be filled in as soon as possible but at the latest Sunday 18:00 and then also be filled with non project related to complete 40 hours.

A week starts at Monday and ends at Sunday. Every entry in a log has to belong to one of the following work-packages: SPMP, SVVP, UTP, ITP, STP, ATP, SCMP, SQAP, URD, SRD, Prototype, Research, ADD, DDD, Code, IT,ST,AT, STD, Formal reviews, Meetings or Presentations.

The PM sends an email to the SM every week, containing the hours spend on the different work packages and the hours spend on following categories: Non project related, General project related, Documentation, specification, design, Source code, Testing, verification, consolidation and rework. Further, for every workpackage, an estimation of remaining hours is added.

3.5 Staffing plan

Name	EMail	Telephone	Function
Lou Somers	lou.somers@oce.com	(040 247) 2805	Senior Management
Mark van den Brand	m.g.j.v.d.brand@tue.nl	(040 247) 2744	Senior Management
Erik Luit	e.j.luit@tue.nl	(040 247) 4338	Technical Advisor
Natalia Sidorova	n.sidorova@tue.nl	(040 247) 3705	Customer
Wilco Belgraver Thissen	w.p.c.belgraver@student.tue.nl	06 44060627	PM
Jelle Hellings	j.a.j.hellings@student.tue.nl	06 13816358	QM
Koen Kivits	j.m.kivits@student.tue.nl	06 30123594	VPM
Nick van der Veeken	n.v.d.veeken@student.tue.nl	06 49864254	VQM
Sander Leemans	s.j.j.leemans@student.tue.nl	0411 683547	CCM
Gijs Direks	g.l.t.direks@student.tue.nl	06 16277201	CM
Roy Berkeveld	r.a.j.berkeveld@student.tue.nl	06 38975366	VCM
Neal van den Eertwegh	n.v.d.eertwegh@student.tue.nl	06 42583346	
Dion Jansen	d.jansen@student.tue.nl	06 22636383	
Kevin van der Pol	k.v.d.pol@student.tue.nl	06 23814445	
Michael van Duijkeren	m.p.v.duijkeren@student.tue.nl	06 41311606	

Chapter 4

Technical process

4.1 Methods, tools and techniques

The methods, tools and techniques used during the course of the project are listed in the SCMP.

4.2 Software documentation

During the project, documents should conform to a number of aspects:

Documents must be of good quality The standards all documents are required to meet are documented in the SCMP[2] with respect to style and in SQAP[3] with respect to content.

Documents must be reviewed The manner in which document reviews are performed is described in the SVVP[4].

The purpose of document reviews is to get docs of high quality The requirements which apply to the approval of documents are given in the SVVP[4].

4.3 Project support functions

Besides Project Management, three other management functions are present. Below a short description of each of them is given. All these management plans are part of the ESA standard[1]. More detailed descriptions can be found there.

Configuration Management The purpose of software configuration management is to plan, organise, control and co-ordinate the identification, storage and change of software through development and transfer. The Configuration Manager writes the SCMP [2] in which plans are outlined for performing these tasks.

Verification and Validation Software Verification and Validation activities check the software against its specifications. In this document one of the team members outlines the plans to perform the verification and validation activities.

Quality Assurance During the project, all documents should be tested against quality standards; both to conform to the chosen project documentation style and to contain correct information of reasonable quality. The regulations and the way in which will be checked whether these are followed are described in the SQAP[3].

Chapter 5

Work packages, schedule, budget

5.1 Work packages

The work packages are defined in the appendix detailing their phase.

5.2 Dependencies

For the UR and the SR phase, a dependency chart is not necessary as dependencies are trivial. The URD must be more or less ready before work on the SRD can start. However, it is unavoidable that, while working on the SRD, new questions arise about the user requirements. So the SRD does not strictly depend on the URD. Likewise working on the ADD can lead to changes to the SRD and so on. The prototype must not be finished later than the SRD since it is part of it. In the AD phase there are few dependencies between tasks. Within the AD phase no work packages are dependent on each other. When the first results of the AD phase are there, the DD phase commences with the first iteration. Work packages don't depend on each other since interfaces between components have been defined in previous phases. The TR phase is the last phase. Obviously, it is not possible to transfer the product before it is ready. Therefore the DD phase must be completed before the TR phase starts.

5.3 Resource requirements

The most important resources during the project are human resources. An overview of resource utilisation during the various project phases is given in section 5.4. Other resources needed include development stations, a server where documents and information can be stored, a printer, network connectivity, a room with sufficient tables and chairs and a telephone suitable for work and meetings. During the project software is required. For example a programming language and a text editor is necessary. The SCMP[2] describes the software that is used during the project.

5.4 Budget and resource allocation

In most projects the only resource is the number of man-hours of the people involved. This is described in the section 5.5, schedule.

5.5 Schedule

This section gives a rough planning of resource usage. The only important resource is man hours.

5.5.1 Deadlines

Management documents approved	October 4th
URD approved	October 4th
Prototype approved	October 27th
SRD approved	November 1st
Intermediate presentation	November 9th
Start first increment	November 9th
Peer review	November 16th
ADD approved	December 14th
DDD approved	December 14th
Coding complete	January 3th
Acceptance test successful	January 8th
Final presentation	January 11th
The project deadline	January 15th

Table 5.1: deadlines

5.5.2 Hour planning per phase

For each phase in the project, an estimate on the amount of resources needed is noted in table 5.2. The time invested by the PM and the technical advisor is not included in this table. These values consist of an initial estimation and can change during the course of this project. In the Progress Reports, updates of this estimation will be given.

5.5.3 Unavailability overview

In the table 5.3 we'll list which of the important people for the project will be unavailable for some period of time, outside exam weeks and holidays.

Phase	**hours**	**Phase end**
UR	400	October 4th
SR	450	November 1st
AD	500	November 29th
DD	1200	January 3rd
TR	150	January 8th
(Margin)	324	
Total	3024	January 15rd

Table 5.2: Hours per phase

Name	Role	Unavailable period
Natalia Sidorova	Customer	-
Erik Luit	Technical advisor	Every Friday
TUE	Facilities	25 December - 1 January
Most members	Members	Study information Days : 16 October, 11 December
Neal, Roy, Kevin, Koen	Members	26-29 October
Jelle Hellings	QM	26 October - 5 November
Dion Jansen	Member	21, 22, 26 October - 2 November
Michael van Duijkeren	Member	16-20 December
Koen Kivits	Member	21-26 December

Table 5.3: Unavailability

Chapter 6

Implementation process

The implementation phase is done incrementally. This influences the way of working with regard to implementation of QIS and the writing of the following documents related to the implementation: ATP, ADD, DDD and SUM. This chapter details the implemention process.

6.1 Requirements on team members

Each team member is required to have completed a Python and Django tutorial. Also, members need to know and apply the coding- and comment standard as specified in the DDD. The adminstration of progress is done using Trac as described in the SCMP[2], therefore team members need to know how to use Trac.

6.2 Iteration delivery

Each week a new so-called *iteration* has to be completed and delivered to the Senior Management. An iteration is a working version of the (unfinished) application, including its documentation. For each iteration a milestone is created in Trac, with associated work packages. Work packages specify groups of tasks that are closely linked together.

6.3 Work packages

A work package specifies tasks related to a set of software requirements that form a relatively self contained unit. A package can be implemented by a development team. A development team consists of at least two people, more is possible. Work packages are described on Trac by means of tickets. The description contains a task list, points of attention and a list of dependencies in relation to other work packages.

6.4 Assignment of work packages

Work package tickets are linked to milestones in Trac. Milestones define when a work package has to be finished. Selecting packages is dones by milestone deadlines. Packages with the earliest milestone deadline have to be finished first. When a work package is finished, the related ticked in Trac is marked as *fixed*. Progress can be monitored using the *Roadmap* in Trac.

When starting on a work package the work package has to be assigned in Trac to a team of developers. The members of the team that are working on it are listed in the field *Assigned to*. The member mentioned first in this list is the team leader. Trac offers the possibility to send emails to a number of people when a task is changed by means of the *CC* field. If there are people involved in a task, but not as developer, their email-addresses must be added in this field.

6.4.1 Work package processing steps

The following list defines the steps of the general procedure for processing a work package, once assigned. In the list, 'the team' refers to 'the team working on this work package'.

1. Detailed design
   (a) Determine which requirements of the URD and SRD should be covered by this work package.
   (b) Design the implementation of the work package. The result of this process should be documented in the ADD according to the design standards mentioned in the ADD.
   (c) Review the design together with group members not part of the team.
2. Implementation and testing
   (a) Implement the work package in a separate branch.
   (b) Test the implementation within the team.
   (c) Verify that the implementation complies with the coding- and comment standard.
   (d) Verify that the implementation complies with the relevant requirements from the URD and SRD.
3. Merging
   (a) Merge the trunk into the branch.
   (b) Test the implementation with members other than those of the team.
   (c) Verify that the merged implementation compiles, works as expected.
   (d) Verify that as a whole, the merged implementation still complies with the URD and SRD.
   (e) If relevant, test the implementation with the client (user testing).
   (f) Merge the branch into the trunk.

(g) Use the comments in the implementation to generate documentation in the DDD.

4. Acceptance and user manual

   (a) Write a number of test cases to verify the implementation of the work package in the ATP.

   (b) Write the user manual for the newly implemented part of the system in the SUM.

6.4.2 Implementation planning

The following table lists the planning for the delivery of iterations and the related work packages. This planning is preliminary and may be adjusted during the implementation phase, depending on the actual progress.

Milestone	Deadline	Related packages
Milestone 1	16-11-2009	Database
		Model manager
Milestone 2	23-11-2009	Organizational
		Employees
		Rights
Milestone 3	30-11-2009	Courses
		Tasks
		Assignments
Milestone 4	7-12-2009	Import data
		Provisional versions
		Reports
Milestone 5	14-12-2009	Closing
		Copying
		Notifications (1)
		GUI
Milestone 6	21-12-2009	NT
		Automatic research hours
		CSV export
		OWIS
		Notifications (2)
		Remarks
Milestone CW		Formulas
		ORCA
		Syllabus+

6.4.3 Work package definitions

The milestones are defined below, together with the dates on which they have to be finished. For the association of work packages and milestones, see Trac. This list is not exhaustive and subject to change.

Database Creating all tables and attributes.

Model manager Adding a model manager, that makes forcing rights easy.

Organizational Including organizational functionality.

Employees Including hires and employee functionality.

Courses Including course instances and education types.

Rights Granting and revoking rights for employees and organizational units.

Tasks Adding and editing other tasks then regular education ones.

Assignments Adding and editing assignments.

Import data Importing test data.

Provisional versions Making courses and workload planning definitive.

Reports Showing reports, possibly by a generator.

Closing Closing a year for a subdepartment.

Copying Copying a year.

Notifications (1) Sending notifications, but not to expertise group leaders.

GUI Updating the GUI and smoothen the experience.

NT Validating login trough the NT system.

Automatic research hours Automatically fill up research hours when required.

CSV export Exporting CSV data.

OWIS Importing and exporting course data to OWIS.

Notifications (2) Sending notifications to expertise group leaders and storing all notifications.

Remarks Adding remarks to some classes.

Formulas Calculating hours for regular course tasks.

ORCA Retrieving data about long leaves from ORCA.

Syllabus+ Sending data about long leaves to Syllabus+.

Appendix A

UR phase

In this phase the main focus is on the definition of the user requirements. Besides the URD the management documents are written during the UR phase.

A.1 Output

The UR phase can be called the problem definition phase. User requirements are documented in the URD, giving the customers view of the problem. The main outputs of the UR phase are the:

- URD
- SPMP including UR and SR appendices
- SCMP including UR and SR appendices
- SQAP including UR and SR appendices
- SVVP including UR and SR appendices
- ATP

A.2 Budget and resource allocation

The table A.1 lists packages for the UR phase as defined in section 5.1 and the budget that has been assigned to each.

Work package	**Budget estimate (man hours)**	**Members**
URD	200	URD team
SPMP	20	PM
SCMP	30	CM/CVM
SQAP	30	QM
SVVP	20	Test team
ATP	30	Test team
Project group meetings	44	Everybody
Initialisation	30	Everybody
Formal reviews	16	QA person and author
Research	30	Protype/Research Team

Table A.1: Budget per work package

Appendix B

SR phase

In the SR phase the main focus is on the definition of the software requirements. Besides SRD the managerial documents are modified for the AD phase.

B.1 Output

The input to the SR phase is the URD and the ESA software engineering standard. The main outputs of the SR phase are the:

- SRD
- Prototype
- SPMP/AD
- SCMP/AD
- SQAP/AD
- SVVP/AD
- STP

B.2 Budget and resource allocation

The table B.1 lists packages for the SR phase as defined in section 5.1 and the budget that has been assigned to each.

Work package	**Budget estimate (man hours)**	**Members**
SRD	250	SRD team
Prototype	70	Protype/Research Team
SPMP	2	PM
SCMP	3	CM/CVM
SQAP	3	QM
SVVP	2	Test team
STP	10	Test team
Project group meetings	44	Everybody
Intermediate presentation	26	Presentation team
Formal reviews	10	QA person and author
Research	30	Protype/Research Team

Table B.1: Budget per work package

Appendix C

AD + DD phase

The AD and DD phase have been merged due to the SM's decision to work with increments. The procedures are described in chapter 6.

C.1 Output

The input to the AD+DD phase is the SRD and the knowledge gained from prototyping. The main outputs of this phase are the:

- ADD
- SPMP/DD
- SCMP/DD
- SQAP/DD
- SVVP/DD
- DDD
- ATP/ITP/UTP
- QIS
- SUM

Work package	**Budget estimate (man hours)**	**Members**
ADD	290	A development team
SPMP	2	PM
SQAP	2	QM
SVVP	2	Test team
DDD	70	A Development team
ITP/UTP	30	Test team
ATP	100	A Development team
SUM	80	A Development team
Code	966	A Development team
Research	30	
Project group meetings	88	Everybody
Formal reviews	20	QA person and author

Table C.1: Budget per work package

Appendix D

TR phase

The main focus of the TR phase is to get the product accepted by and delivered to the customer.

D.1 Output

The input to the SR phase is the product. The main output of the TR phase is the STD

Work package	**Budget estimate (man hours)**	**Members**
STD	20	Test team
IT, ST, AT	80	
Project group meetings	10	Everybody
Final presentation	40	Presentation team 2

Table D.1: Budget per work package