Software Project Management Plan

for the

Project Arena

Revision: 1.11 Date: 2002/11/21 16:09:41

Contents

1 Introduction

1.1 Project overview

With the arrival of the global broadband network infrastructure, new game concepts are possible and resulted in the creation of a wealth of Multiplayer Online Games (MOGs), for example, Quake, CounterStrike, Ultima Online, EverQuest, or WarCraft. These games share common concepts:

- A client-/server architecture, with one dedicated server with control over player movement and world objects.
- Extensible, but pre-designed sets of maps, objects, and weapons. Typically, these game elements are created by graphic designers and artists in substantial manual effort.
- Freedom of player movement and the ability to manipulate world objects.

But after the broadcasting infrastructure of radio and television was supplemented by two-way network connections, another use of the technology was enabled: peer-to-peer (P2P) networks. Thus far they are mainly used for one-on-one messaging or simple file exchange.

The newly available FRAG[1] framework allows to go one step further. FRAG manages peer-to-peer communication, distributed object synchronization and message transport

[1] Framework for Realtime Ad-hoc Games

for object-based 2D or 3D game worlds. FRAG is part of the Global$^{S}_{E}$ project ARENA providing an infrastructure for tournaments in mobile environments.

This project will develop a peer-to-peer multiplayer online game on top of the FRAG framework. A functional prototype will be demonstrated at the end of the project, which is based on the technologies and infrastructure available.

1.2 Project deliverables

The project will a produce a running system that allows multiple players to play a real-time game over an ad-hoc network. The following items will be produced by the ARENA System:

- A **Software Project Management Plan** defining the technical and managerial processes necessary for the development and delivery of the ARENA system (This document)
- **Agreement** between client and developers, representing a contract between the client and the developers of what is going to be delivered.
- A **Requirements Analysis Document** describing the functional and global requirements of the system as well as 4 models - the use case model, the object model, the functional model and the dynamic model. This document is created in interaction with the application domain experts.
- A **System Design Document** describing the design goals, tradeoffs made between design goals, the high level decomposition of the system, concurrency identification, hardware/software platforms, data management, global resource handling, software control implementation and boundary conditions. This document forms the basis of the object design. This document is read by the analyst as well as the object designer.
- An **Object Design Document**, which is composed of two documents. The first document is an updated RAD. The code related data will be in the form of JavaDoc output from the code from each team.
- A **Test Manual** describing the unit and system tests performed on the ARENA system before delivery along with expected and actual results. This document is used by the developers and maintainers.
- **Source code** for all subsystems of the ARENA System.

The ARENA System documentation will describe the principles of operation. The delivery consists of a presentation of the system, a demonstration of the working system and the successful passing of the acceptance test. The client expects the acceptance test to be successfully demonstrated on 22 Jan 2003 in room 01.07.014. All work deliverables will be provided online on a project homepage. The work products will also be delivered on a DVD, 31 Jan 2003.

1.3 Evolution of this document

The software project management plan is under version control. Proposed changes and new versions of the plan are announced on the Management Issues BBoard and are made available to all the project members.

1.4 References

References

[BD00] Bernd Bruegge and Allen H. Dutoit. *Object-Oriented Software Engineering: Conquering Complex and Changing Systems.* Prentice Hall, 2000.

[GHJV95] Eric Gamma, Richard Helm, Ralph Johnson, and John Vlissides, editors. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley Longman Limited Harlow, England Reading, Massachusetts Menlo Park, California New York, 1995.

[JCJÖ94] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Övergaard. *Object-Oriented Software Engineering – A Use Case Driven Approach.* Addison Wesley Longman Limited Harlow, England Reading, Massachusetts Menlo Park, California New York, 1994.

[IEEE 828] IEEE Standard for Software Configuration Management Plans, ANSI/IEEE Std. 828-199.

[IEEE 1058] IEEE Standard for Software Project Management ANSI/IEEEStd.1058.1-1987.

[IEEE 1074] IEEE Standard for Developing Software Life Cycle Processes, ANSI/IEEE Std. 1074-1991.

1.5 Definitions and acronyms

API	Application Programming Interface
CASE	Computer Aided Software Engineering
CVS	Concurrent Versions System
GUI	Graphical User Interface
HCI	Human Computer Interaction
JDK	Java Development Kit
ODD	Object Design Document
OMT	Object-Oriented Modeling Technique
RAD	Requirements Analysis Document
SDD	System Design Document
SPMP	Software Project Management Plan
TM	Test Manual
UML	Unified Modeling Language

2 Project organization

This section describes the project organization in a process model, organizational model, and a list of project responsibilities.

2.1 Process model

The project is initiated on 14 Oct 2002 and terminated with the end of the semester on 7 Feb 2003. Major milestones are the Client Project Review on 10 Jan 2003 in room 01.07.014 and the Client Acceptance Test on 22 Jan 2003 in room 01.07.014.

The project uses an object-oriented design methodology based on the Objectory lifecycle process and uses UML for the development of the software. The development process is organized in several activities. The members of the project are organized in teams. At the end of each activity up to and including testing, each team submits documents describing the achievement of the activity. The individual approved documents produced by the teams are considered work products and are part of the software documentation. The team documents are under version control using CVS running on a Mac OS X platform. Links to the team documentation are available from the team homepages and the course electronic bulletin boards. The links to the major documents on the CVS server are also available from the project home page. The activities and milestones are described in the next following sections.

2.1.1 Project Planning

Project planning includes description of project tasks, activities and functions, dependencies, resource requirements and a detailed schedule. This activity results in the software project management plan for the ARENA System. Another output of the planning phase is the project agreement, which is issued after the design activity is completed.

2.1.2 Requirements Analysis

The requirements analysis activity takes the problem statement and reviews it in terms of consistency, completeness and feasibility. During this activity, a set of models of the proposed system is determined by interacting with the clients resulting in the requirements model. The main part of the requirements model are four models: the use case model describing the complete functionality of the system, the object model, the functional model and the dynamic model.

2.1.3 Analysis Review

Review of software project management plan, requirements analysis and design. The meeting will take place on 6 Nov 2002 in room 01.07.014. The Analysis Review consists of a set of presentations given by members of the ARENA project. Project Management will review these slides and post their comments on the project discuss BBoard.

2.1.4 System Design

The purpose of the system design activity is to devise a system architecture that maps the analysis model to the chosen target environment. The major part of the system design phase is the design of subsystems, that is, the decomposition of the system with respect to the chosen target platform. The system design activity also refines the use cases from the analysis model and describes in terms of interaction diagrams how the objects interact in each specific use case.

2.1.5 Client Project Review

Review of project plan, requirements analysis and design decisions. The client liaison will be present at the meeting. The meeting will take place on 10 Jan 2003 in room 01.07.014. The Client Project Review presentation slides will be made available to the client.

2.1.6 Object Design Phase

The object design phase specifies the fully typed API for each subsystem. New classes are added to the analysis object model if necessitated by the system architecture. Attributes and methods for each object are fully typed.

2.1.7 Functional Prototype Demonstration

This activity involves successful execution of a functional prototype of the ARENA System using stubs. The functional prototype of the ARENA system will be presented during the internal review on 15 Jan 2003 in room 01.07.014.

2.1.8 System Integration Prototype Demonstration

This activity involves the demonstration of a fully functional system prototype based on the subsystem decomposition. Each subsystem is represented by its service. All service operations can be called by other subsystems using Java method invocation. The implementation of the services can be stubbed out.

2.1.9 Implementation

The focus of this activity is on coding the individual objects described in the object design document.

2.1.10 Unit Testing

During unit testing, test suites are designed and executed for objects or collections of objects in each subsystem. Unit testing enables the individual subsystems to be tested independent from the status of the other subsystems. The result of this activity is part of the test manual that describes how to operate the test suite and how to interpret the test results.

2.1.11 System Integration

During this activity an integration strategy is devised, that specifies the order in which the subsystems of the ARENA system are integrated and tested with respect to the use

cases defined in the analysis model. The system integration strategy and the subsystem tests are described in the Test Manual.

2.1.12 System Testing

Structural Testing: This activity tests the major data paths in the complete ARENA System. Functional Testing: Tests the major functionality (use cases) with the complete ARENA System. The basis for the functional testing activity is the test manual which is revised according to the results of the system testing phase. Alpha-test (Client Acceptance Test): The system is tested to make sure it passes the client acceptance criteria as defined in the project agreement.

2.1.13 Manual Integration

During this activity, the project deliverables are revised. As a result, a complete set of documents consisting of the software project management plan, requirements analysis document, software design document, test manual and source code is made available on the project home page. The system documentation will also be printed on a DVD. Each of the students taking the course as well as the clients will receive a DVD.

2.1.14 Client Acceptance Test

At the Client Acceptance Test, a slide presentation, scenario film showing and software demonstration will be given to the clients. The software developed during the project will be demonstrated. The clients will attend the client acceptance test in person or via video conference.

2.2 Organizational structure

Figure 1 on the following page shows an organizational chart of the people involved in the development of the ARENA system. This role chart is meant as an overview of cross-functional roles. The roles will be taken by students, who will be coached by the assigned coaches. The diagram does not show the developer team distribution.

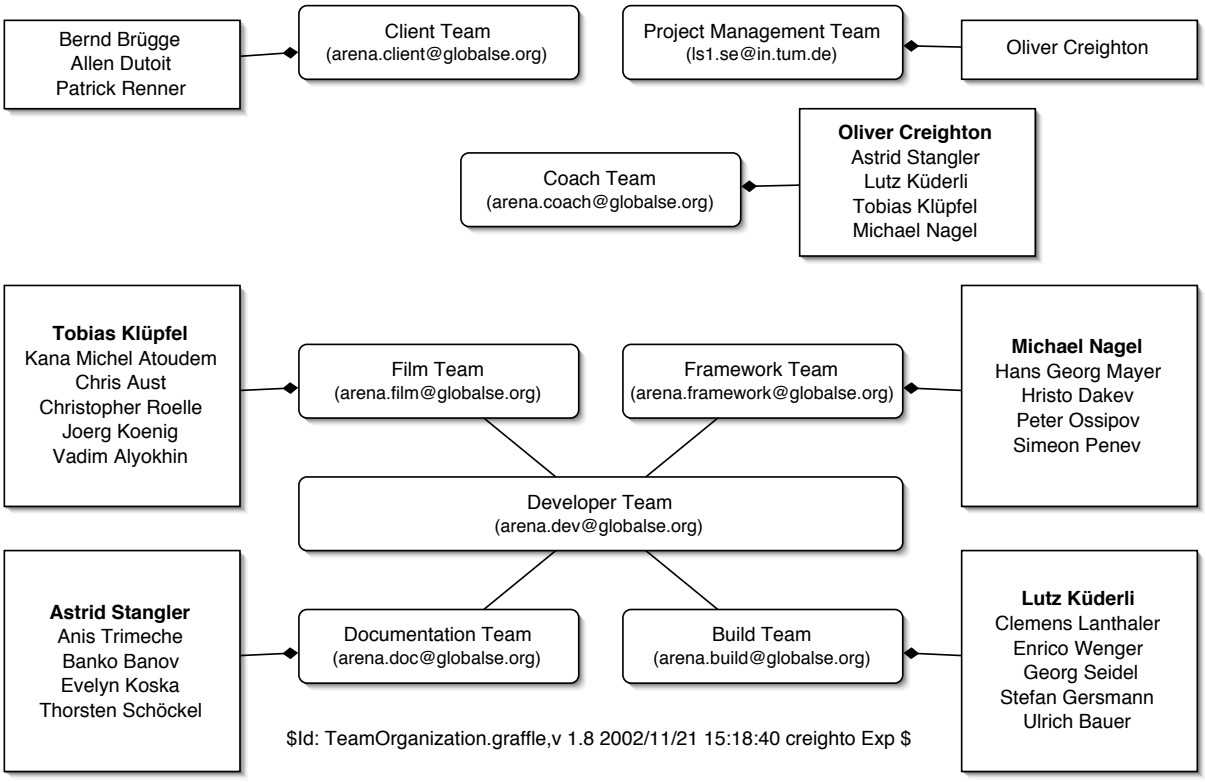

Figure 1: The Organizational Structure of Arena

2.2.1 Teams and Tasks

The project is organized in teams, which all have specific roles in the project. These roles are:

- **Client Team:** speaks for a yet-to-be-identified target customer base. This role is responsible for providing feedback on the system under development and can be consulted for requirements elicitation.
- **Project Management Team:** responsible for the project organization. This role helps resolving management and design issues.
- **Coach Team:** provides team leaders feedback on how they organize and manage their teams. This role is attending developer team meetings and has as second responsibility the role of moderator in cross-functional team meetings.

- **Film Team:** cross-functional team who produces (scripts, plans, films, edits, and delivers) video material to be used in the Client Acceptance Test and on the project DVD.
- **Framework Team:** cross-functional team who negotiates and decides changes in the ARENA architecture and FRAG framework. This role is responsible for finding common solutions that should be added to the framework.
- **Documentation Team:** cross-functional team who negotiates documentation standards and oversees the completion of all required documentation. This role is responsible for managing the documentation process of the entire project.
- **Build Team:** cross-functional team who negotiates and implements the build and testing environment. This role is responsible for the integrated development environment and maintenance of the build infrastructure of the entire project.
- **Developer Team:** the development teams who will be assigned to deliver work products according to the work breakdown structure. The development teams will be reassembled dynamically, depending on the tasks and action items that need to be done. Figure 2 shows the current setup of the developer teams.

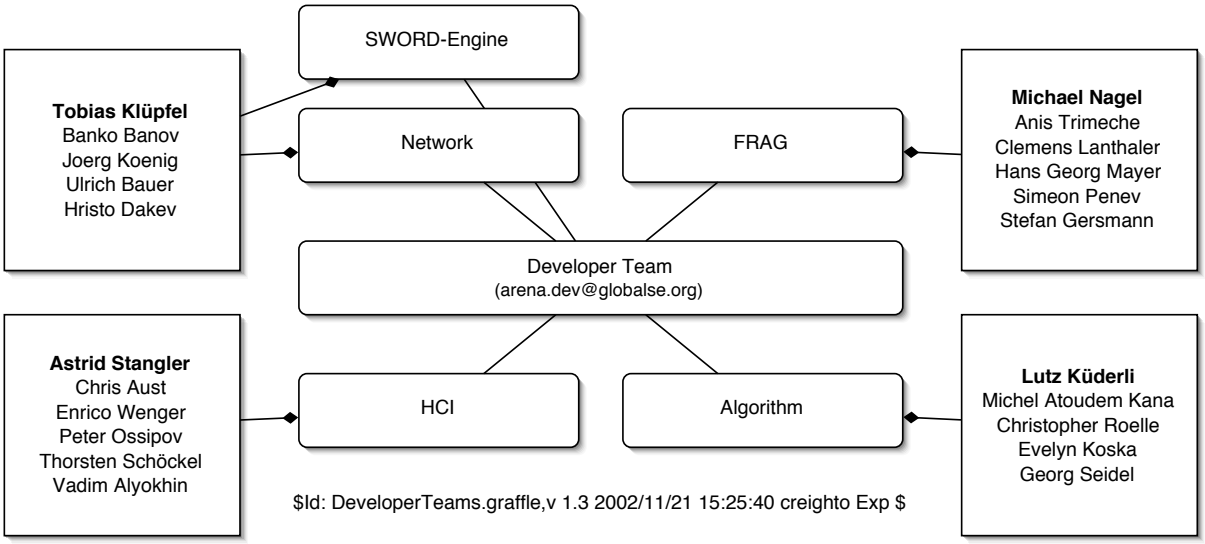

Figure 2: The ARENA Developer Teams

Figure 3 on the following page shows the work breakdown structure for the ARENA project. The containers are work areas that have been identified and contain tasks which are assigned to different developer teams.

The current assignment of tasks for the developer teams are:

Network: become Zeroconf experts (Rendezvous implementation).
Work Products: discover peers, network initialization protocol

FRAG: become FRAG experts.
Work Products: draw FRAG objects, draw World

HCI: design GUI
Work Products: create 3 mockups, identify and integrate new I/O device

Algorithm: invent and implement world generation algorithm
Work Products: generate 3 different worlds, define mission description language, generate missions algorithmically

SWORD-Engine: plan, design, and implement the game engine
Work Products: Adventurer and Item classes with properties, game integration platform

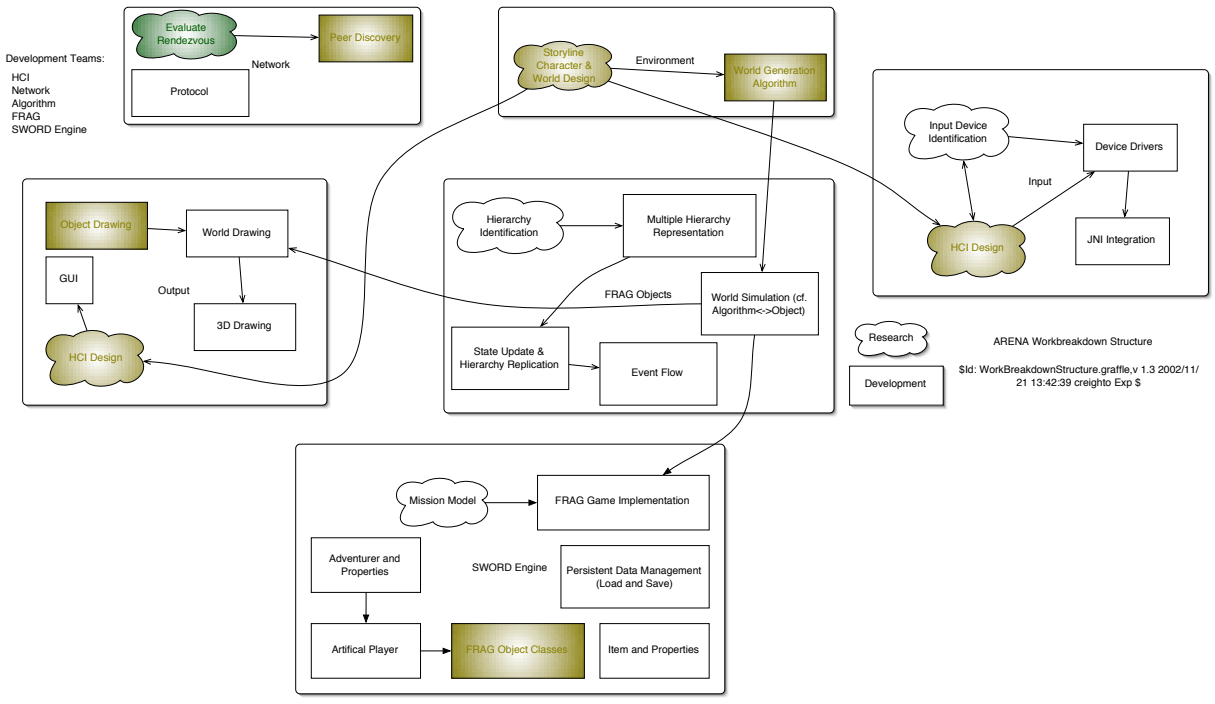

Figure 3: The ARENA Work Breakdown Structure

2.3 Organizational boundaries and interfaces

This section describes the two main forms of organizational communication: The electronic BBoards and the scheduled meetings.

Title	Topic
Announcements	Project announcements
Management Issues	Schedule, risks, and management decisions
Client	Primary forum for interchange with the clients
Coaches	Developer coaching and tutorial authoring
Homework	Electronic submission of homework
Off-topic Discussion	For experimenting with the BBoard functionality
Framework	Software architecture and system/framework boundary
Documentation	Documentation standards and review processes
Film	Scenario film and its production
Build	Coding standards and build and testing environment
Algorithm	Game world generation algorithm
FRAG	FRAG framework development
HCI	User Interface and I/O devices
Network	Peer-to-Peer networking component
SWORD-Engine	Game Engine

Table 1: Electronic BBoards for the ARENA Project

2.3.1 Electronic BBoard Communication

The Lotus Notes Databases shown in Table 1 will be used for electronic communication in the ARENA project. These BBoards provide for a structured discussion using a rhetorical model based on Issues, Proposals, Arguments, and Resolutions.

Every team member has to:

- Bookmark the Announcements, Issues, and Client BBoard
- Bookmark the team specific BBoards
- Check these BBoards at least twice a day

Communication with the client is primarily via the Client BBoard. As the need arises direct e-mail and/or telephone contact is set up with specific consultants within the client organization.

2.3.2 Meeting Times

There is a weekly project meeting for each group. The time and location can be seen in Table 2. There is a weekly all-hands meeting Wednesdays at 16:00 in room 01.07.014.

Team	Day of the week	Time	Room
Coaches	Tuesday	16:00	01.07.058
Framework	Monday	14:00	01.07.011b
Documentation	Thursday	14:45	01.07.058
Film	Thursday	19:00	Munich downtown
Build	Thursday	17:00	01.07.058
Algorithm	Monday	18:00	01.07.058
FRAG	Wednesday	11:00	01.07.058
HCI	Thursday	13:00	01.07.058
Network	Tuesday	18:00	01.07.014
SWORD-Engine	Tuesday	18:00	01.07.014

Table 2: Meeting Times and Locations for the ARENA Project

2.4 Project responsibilities

Management of the ARENA System is done with the following roles: project management, coach, framework liaison, documentation liaison, build liaison, and film liaison.

2.4.1 Project Management

The project management function has the following responsibilities:

- Integrate project plan and schedule
- Assign presentations (project meetings, client reviews, client acceptance test) to project members
- Schedule meetings with clients
- Listening to gripes from the team members
- Resolve conflicts if they cannot be resolved otherwise

2.4.2 Coach

The coach has the following responsibilities:

- Review weekly team progress

- Attend weekly team meetings
- Insist that guidelines are followed

2.4.3 Framework Liaison

The liaison interacts with the liaisons of the other teams and with the project management. Each developer team has a liaison to the Framework Team. The responsibilities of the liaison are:

- Make available public definitions of each subsystem service (its API) to the other teams (ensure consistency, etc.)
- Coordinate tasks that overlap subsystems with the teams
- Responsible for team negotiations, that is, resolve technical issues spanning more than one subsystem
- Defines the software architecture for Arena
- Defines the class library and framework boundary of Arena

2.4.4 Documentation Liaison

The liaison in each team is responsible for producing the documentation of the current project phase and:

- Define documentation standards and monitor adherence
- Collect, proofread and distribute team documentation

2.4.5 Build Liaison

The responsibilities of the build liaison in each team are:

- Define and implement build process and monitor nightly builds
- Coordinate change requests
- Provide version control for group's working directory
- Coordinates configuration management issues with other groups
- Installation of group specific software and hardware

2.4.6 Film Liaison

The responsibilities of the film liaison in each team are:

- Identify the important work products of the team which have to be in the film
- Deliver good visualizations for the film
- Coordinate film with other documentation

3 Managerial process

This section describes the managerial process in objectives, priorities, assumptions, dependencies, and constraints. It explains the contingency plans for identified risks and the installed monitoring and controlling mechanisms.

3.1 Management objectives and priorities

The philosophy of this project is to provide a vehicle for students to get hands-on experience with the technical and managerial aspects of a complex software problem. The emphasis is on team work and encouraging individual teams to work together towards the goal of implementing the ARENA system completely.

3.2 Assumptions, dependencies and constraints

The functionality of the ARENA System is achieved when the client acceptance test can be executed. Each software development activity results in one or more documents to be submitted to the project management before the deadline. Each document is reviewed at least once by the project management before it is accepted and becomes a baseline document. The following documents will be graded: SPMP, RAD, SDD, ODD, and TM. The agenda, minutes, action items and issues for each weekly team meeting have to be posted. The complete set of these is also required. We will give a "Schein" to everybody who actively participates in the project, if all the project deliverables are delivered and the ARENA system passes the client acceptance test as defined in the requirements analysis document. The ARENA System is a project that puts emphasis on collaboration, not competition between the students. We will not accept a system that is done by one team alone.

3.2.1 Assumptions

To be filled in by the individual teams.

3.2.2 Dependencies

To be filled in by the individual developer teams.

- Algorithm:
- HCI:
- Network:
- FRAG:
- SWORD-Engine:

3.2.3 Constraints

- Language. The system will be programmed in Java, except for performance critical or hardware-near (device driver) elements, which may be written in Objective C, if so negotiated with project management. Deviation from using the Java language need explicit agreement from the client.
- Framework. The system will be built in such a way that it uses components of the FRAG framework, and components developed in the project should themselves be useable in other FRAG systems.
- Platform. The system needs to be demonstrated on the provided hard- and software platform: Apple iBooks running Mac OS X v10.2.

3.3 Risk management

To be filled in by the individual teams from submissions to the Management Issues BBoard.

3.4 Monitoring and controlling mechanisms

For each project meeting each team produces an agenda and the minutes of the meeting. The minutes have to contain explicitly the action items assigned during the meeting. The agenda and minutes are posted on team specific bulletin boards by the minute taker of the meeting. The baseline documents are reviewed by the project management. It is expected that each document undergoes one or more iterations.

4 Technical process

Provide more information as necessary.

4.1 Methods, tools, and techniques

Our development methodology is based on a combination of use cases (from the OOSE methodology) [JCJÖ94] combined with the OMT methodology. The following tools are available to support the management and development of the Arena project:

- Apple Developer Tools (Project Builder, cvs, JDK)
- Apple iTools: iCal, Address Book, iMovie, iDVD
- Together Control Center
- OmniGraffle
- Microsoft Office X Powerpoint
- TeXShop
- Final Cut Pro
- DVD Studio Pro

4.2 Software documentation

The following activities result in a project deliverable:

- Requirements Analysis: Requirements Analysis Document (RAD)

- Analysis Review: Analysis Review Slides and Scenario Film
- Risk Analysis: Software Project Management Plan (SPMP)
- System Design: System Design Document (SDD)
- Client Review: Client Review Slides and Video
- Object Design: Object Design Document (ODD)
- Reviews: Review Presentation Slides
- Implementation and Unit Testing: Code
- System Integration and System Testing: Test Manual
- Delivery: Client Acceptance Test Slides and Video

4.3 Project support functions

Provide more information as necessary.

4.4 Work elements, schedule, and budget

Provide more information as necessary.

4.4.1 Overall Project Plan

The overall project plan follows the sawtooth model, a modified waterfall model. 3 prototypes have to be delivered: A graphical user interface, a functional prototype and a system integration prototype. Analysis is started before Project Planning is finished. System Design is followed by Object Design. Important Milestones are the Analysis Review on 6 Nov 2002 in room 01.07.014, the Project Review on 10 Jan 2003 in room 01.07.014 and the Object Design Review on 15 Jan 2003 in room 01.07.014. Implementation and Unit Testing are scheduled to overlap significantly. System Integration is scheduled to immediately follow Unit Testing. System Testing starts immediately after system integration and leads to the Client Acceptance Test on 22 Jan 2003 in room 01.07.014.

4.4.2 Schedule

31 Oct 2002 **Analysis Complete: *The Halloween Document.*** By this date the Requirements Analysis Document (RAD) is baselined and the review process starts. Part of the RAD is a script and storyboard for the DVD (the Scenarios).

6 Nov 2002 **Requirements Analysis Review.** By this date the developers have gotten feedback on the RAD and give a presentation to the client about the analysis of the requirements.

15 Nov 2002 **Design Complete.** By this date the System Design Document (SDD) is baselined and the review process starts.

20 Nov 2002 **System Design Review.** By this date the developers have gotten feedback on the SDD and give a presentation to the other developers (and possibly the client) about the system design.

29 Nov 2002 **APIs Complete.** By this date the Object Design Document (ODD), describing the APIs of the subsystems, is baselined and the review process starts.

4 Dec 2002 **Unit Test.** By this date the developers have gotten feedback on the ODD and API quality and give a presentation to developers about the use of the APIs and how they can be tested automatically.

13 Dec 2002 **Test Drivers and Stubs Complete.** By this date every subsystem is implemented enough for other components to compile and run.

18 Dec 2002 **Integration Test Suite Presentation.** This presentation focuses on how the integration tests are conducted and what is expected of the subsystems.

10 Jan 2003 **Client Acceptance Test Preparation Done.** By this date the DVD that will be used during the Client Acceptance Test (CAT) is complete. Also, the presentation outline and demonstration script is finalized. This is where the review process for the presentation starts.

15 Jan 2003 **Client Acceptance Test Dry-Run.** By this date the developers have gotten feedback on their planned presentation, and conduct a dress-rehearsal of the CAT. After that, the developers get feedback on what to focus on in the actual CAT.

22 Jan 2003 **Client Acceptance Test.** This presentation explains all details of the system to the clients, and is filmed for a showcase DVD of the system.

31 Jan 2003 **Documentation Complete.** By this date all developer documentation is completed and reviewed.

7 Feb 2003 **Process Analysis Complete.** By this date voluntary focus groups are formed and interviewed. A process analysis document is prepared and given to the developers for review.

19 Feb 2003 **Project Archived and Structures Consolidated.** By this date everything that regards the project is consolidated into a project archive, consisting of all project work products, reviews, and deliverables. The last step is the deletion of all temporary data and the closure of the project infrastructure (project "lights out").

4.4.3 Team plans

Provide more information as necessary.