

FUNKY TORRENTS

©The Group

Software Project Management Plan

Version 1.0
IT University of Göteborg
TIG029 – Software Architecture for Distributed Systems

Revision History

Date	Version	Description	Author
28.09.2010	0.1	First draft created	Savas Aydin
30.09.2010	0.2	Updated/Changed Part 4 and 5 considering supervisor's feedback	Savas Aydin
04.10.2010	0.4	Updated Schedule plan and added product backlog	Savas Aydin
28.10.2010	0.8	Updated schedule and risk plan after group discussion	The Group
02.12.2010	0.9	Updated schedule plan and added detailed Gantt chart.	Savas Aydin
21.12.2010	1.0	Final review	Savas Aydin

Table 1. Change history of SPMP

Preface

This Software Project Management Plan describes project plan of course TIG029, Software Architecture for Distributed Systems, at IT University to be taken by The Group. This paper aims to document the agreed deliverables and dates and ensure that requirements fulfilled. Besides, it is intended to be used by supervisors and teachers responsible for the course as delivery of the software project documentation to evaluate and grade project manager of The Group.

Table of Content

This template is originated considering other SPMP templates. ^[1, 2]

Revision History	2
Preface.....	3
Table of Content	4
1. Introduction.....	5
1.1. Project Overview.....	5
1.1.1. Purpose, Scope and Objectives	5
1.1.2. Assumptions and Constraints.....	5
1.1.3. Project Deliverables	5
1.2. Evolution of the plan.....	6
2. Definitions and Acronyms	6
3. Project Organization	7
3.1. External Interfaces.....	7
3.2. Internal Interfaces.....	7
3.3. Roles and Responsibilities	7
3.4. Tools, Techniques and Methodology	8
4. Project Management Plan	9
4.1. Schedule Plan	9
4.2. Risk Management.....	11
5. References.....	13
6. Attachment.....	14

1. Introduction

1.1. Project Overview

1.1.1. Purpose, Scope and Objectives

The purpose of this project is to make a peer-to-peer solution for a local public library which is extending into online distribution of audio books since the standard solution may cause problems when distributing large amount of data.⁴

The SPMP is only associated with Funky Torrent project. Even though the project is given as part of university course where different project groups get the same task, each group is free to interpret the task as they wish.

The Group aims to develop BitTorrent client program which is designed for sharing data by reading torrent-file and communicated with other peers.

1.1.2. Assumptions and Constraints

Funky Torrents is planned based on the assumption of which initial estimations in SPMP extends/shrinks %30. On the other hand, Funky Torrents is limited with time, staff and success requirements, but cost. There is no experienced staff in The Group. The project has almost 3 months to develop working prototype product. The Group is not familiar to Erlang which is main programming language. Project provides each member to take on one of specific task. (See Gantt chart for scheduling and course PM for success requirements.)

1.1.3. Project Deliverables

Working software prototype, its source code and documentation of project that is listed as following:

- SPMP, responsibility of project manager
- SRS & SAD, responsibility of software architect.
- SDS, responsibility of designer.
- STP & STC & STP, responsibility of quality manager.

1.2. Evolution of the plan

Project manager started planning on 27th of September and will take almost 6 working days to define and analyze objectives, framework for project, and schedule it. The Group has appropriate tools, techniques and methodology to be more flexible for changes/updates. (See part 3.4 for further information)

2. Definitions and Acronyms

- BitTorrent – It is a data sharing protocol by peer-to-peer to distribute large amount of files.³
- Erlang (programming language) – It is a concurrent programming language developed by Ericsson to support distributed, and fault tolerant applications.⁵
- Funky Torrents – Torrent client product developed by The Group
- Scrum – It is an iterative method mostly used in agile software development.⁶
- The Group – The project group developing the Funky Torrents.
- TIG029 – The code of lecture of given project, Project: Software Architecture for Distributed Systems
- Torrent-file – It stores data used in BitTorrent and contains the URLs of multiple trackers.³
- DCD – Design class diagram
- QM – Quality Management
- RUP – Rational Unified Process
- SAD – Software Architecture Description
- SDS – Software Design Specification
- SPMP – Software Project Management Plan
- SRS – Software Requirement Specification
- STC - Software Test Cases
- STP - Software Test Plan
- STR - Software Test Report
- TAD – Technical Analysis and Design
- TBD – To be determined
- UML – Unified Modeling Language
- WBS – Work Breakdown Structure

3. Project Organization

3.1. External Interfaces

This project is carried out under the observation of responsible teachers from the course, Hans Svensson and Ulrik Eklund, and supervisor, Magnus Klaar.

3.2. Internal Interfaces

The Group consists of following members:

- Savas Aydin – Project Manager
- Magnus Bergqvist – Software Architect
- Michal Musialik – Designer
- Ali Issa – Quality Manager
- Aydan Beydzhanov Halilov – User Interface Designer
- Björn Eriksson – Communications Expert

3.3. Roles and Responsibilities

Team Role	Responsibilities
Project Manager	<ul style="list-style-type: none">➤ Hold and chair the meetings➤ Measure progress and act pro-actively if planning is in danger➤ Drive project planning and use iterative software process➤ Ensure that enough time is left for testing and writing of the report➤ Create a risk estimation and follow-up
Software Architect	<ul style="list-style-type: none">➤ Main driver to establish a software architecture➤ Monitors that architecture is respected ,or that diversions from architecture are well documented and communicated in the team➤ Make sure interfaces are clearly specified and that designer and tester have access to them, also main responsible for integration.➤ Responsible for the non-functional requirement

Designer	<ul style="list-style-type: none"> ➤ Main driver for software design ➤ Ensure balance between spending time on design documents and actual programming ➤ Set-up coding infra-structure, such as SVN repository, directories, scripts
Quality Manager	<ul style="list-style-type: none"> ➤ Lead the software quality efforts, in particular testing. ➤ Main driver of test case design, focus on quality of code, not quality of development process ➤ Test case work is started as soon as interface description and low-level design are produced ➤ Test infra-structure, such as automated test suite and BT tracker ➤ Responsible for coding standards within the project
User Interface Designer	<ul style="list-style-type: none"> ➤ Design the user interface ➤ Justify the user interface design, with experiments and/or theory
Communications Experts	<ul style="list-style-type: none"> ➤ Study the BitTorrent protocol ➤ Investigate common BitTorrent features

Table 2. Roles and Responsibilities in The Group⁽⁷⁾

3.4.Tools, Techniques and Methodology

Tool	Purpose	Acquired
GNU Emacs 23	Text editor which has erlang plug-in.	www.gnu.org/software/emacs/
Erlang/OTP R13B04	Main programming language	http://www.erlang.org/download.html
SVN/Drop box	File sharing	www.dropbox.com
Photoshop/GIMP	Graphic Tool	www.gimp.com
IBM Rational ClearQuest	Defect Tracking system	QM
UML 2.0	Modeling Language for DCD	TAD

Table 3. Tools and techniques used in Funky Torrents

RUP methodology is used in this project. It is an iterative software developing process. It gives the team a strict approach to spiting up tasks and responsibilities through developing team. Its purpose is to ensure high-end software that meets requirements and expectation of its users

within the determined time schedule. Besides, it aims to make the process flexible against changes in functional requirements.

4. Project Management Plan

4.1. Schedule Plan

Project calendar is created considering by project milestones, and other lectures:

- Under normal circumstances the working hours for "The Group" determined as 4 hours each day 5 days in a week. (Monday - Friday: 10:00 - 12:00 & 13:00 - 15:00)
- Exception week; The Group decided to have a week off to study software architecture examination. (15.11.2010 - 24.11.2010)
- Intensive work weeks; The Group decided to work on project full time includes weekends, starts after 3rd sprint to success the project. (03.12.2010 - 17.12.2010, Monday - Friday: 10:00 - 12:00 & 13:00 - 17:00, Weekends: 13:00 - 15:00 & 16:00 - 19:00)

The Group has regular group meetings every Monday, Wednesday and Friday, and meetings with responsible supervisor in other word, external stakeholder, every Friday but 08.10.2010 due to intensive scheduled lectures and exception week on project calendar.

Scrum is part of regular group meetings, starts in the beginning and takes approximately half an hour. The content of scrums will be not detailed as required in software project since it is not a large-scale project. Common topics will be answers to following questions:

- What have you done since last meeting?
- What will you do next?
- Do you have problem with that?

All related with project welcome to daily scrum meetings but during the session, only people have roles in scrum can speak.⁶ The meeting should start at 10 in the morning and same place, room CP/M in 2nd year students square, and starts exactly on time.

Fridays after lunch, The Group sits to meet with supervisor. Basically supervisors are previous year students who took this project before and can share their experiment with new project

groups. Besides, they are mediators to deliver team problems/questions about project process to the teacher, external primary stakeholder. It saves time, thereby accelerates the project process.

Project manager has determined some control points, consists of 3 parts, called Sprint. Each sprint is very important milestone to track schedule of project and has different requirements for both code and documentation as listed in the table 4. Besides, there are 5 deadline has already determined in course plan.

The aim of each sprint is to identify how much of work has been done or not completed.⁶ The Group is expecting all stakeholders, at least responsible supervisor of Funky Torrents, to involve the sprint meetings to present completed work within 2 hours time limit.

Use stories and tasks are detailed in product backlog.

Milestone	Date	Goal
Initial SPMP	05.10.2010	Initial plan of Funky Torrents
Sprint 1	19.10.2010	Handle the complexity of the project by understanding, analyzing and designing requirements
Sprint 2	08.11.2010	Start coding
Sprint 3 (Release 1)	03.12.2010	Have necessary modules to download (Before Implementation)
Final Sprint (Release 2)	17.12.2010	Implement all written codes on functional web interface (Working prototype)
Documentation submission	22.12.2010	Hand-in final versions of all necessary documents.
Reflection documentation	07.01.2010	Self and group evaluation
Oral presentation	13.01.2010	

Table 4. Schedule of Milestone plan.

This table is very simply explanation of the logic under Sprints. Moreover, there are some exceptions, i.e. hand in initial version of SPMP is not the same time with other reports and submission of source code is almost a week earlier then other document submission, October 5th.

However, detailed schedule information of each tasks and assigned resources for those tasks are in TheGroup_GanttChart.pdf.

As you can see at table 4, The Group has 8 scheduled milestones respectively;

- Hand-in initial version of SPMP, 05.10.2010
- 1st Sprint, 19.10.2010
- 2nd Sprint, 08.11.2010
- 3rd Sprint, 03.12.2010
- Final Sprint, 17.12.2010
- Entire Documentation Submission, 22.12.2010
- Reflection documentation, 07.01.2010
- Oral Presentation, 13.01.2010

Project Name	Funky Torrents
Company Name	The Group
Project Start	9/28/2010 8:00:00 AM
Project Finish	1/13/2011 05:00:00 PM
Current Date	21/12/2010

Table 5. General information about Funky Torrents

Funky Torrents starts and ends on given dates in table 5.

4.2. Risk Management

Early in the project, project manager with developers defined the possible risks that could slow the project down or cause it to fail. Afterwards, it has been given them points from 1 to 5. (1 refers low, 3 medium and 5 high). Team has to consider about both impact of the potential risk to the project and likelihood of occurrence of them. Then the result of fact that how much those potential risks affect project can be computed by multiplying likelihood and impact of the risks.

Nature of Risk	Likelihood	Impact	Score	Actions required and who will take responsibility to manage the risk
Erlang learning	2	5	10	➤ Study and exercise more on Erlang programming language
Sickness	1	2	2	➤ Share his/her tasks among group
Motivation	3	4	12	➤ Be positive and see happy future
Communication	2	4	8	➤ Arrange activities together
Technical Problems	1	3	3	➤ Find out necessary and appropriate tools/techniques
Software				➤ Read/search more about it.
Architecture learning	2	4	8	
WBS	3	4	12	➤ Reallocate the roles considering capabilities of members
Losing team member	1	2	2	➤ Share his/her part among group.
BitTorrent protocol learning	2	4	8	➤ The Group members are responsible for learning the protocol by searching and sharing ideas.

Table 6. Risk Analysis

Table 6 indicates our potential unexpected events, their effects to Funky Torrents and the required actions against those risks.

Either motivation or WBS protocol learning is most vital risk even though neither of them has the highest score in terms of impact and likelihood but total score. Early in the project, members assigned roles that believe it is the best match based on their capabilities. The reason of fact that it has one of the highest risk degrees is that The Group has almost no experience on tasks but theoretical knowledge. The Group agreed on to reallocate the roles among group until Sprint 1, since first sprint gives the group great opportunity to understand if everything is right in the distribution of task. Motivation becomes problematic issue if the things get out of control. The Group has some activities to keep motivation up such as having lunch together and meeting

outside of office to talk except project. Those examples as well provide a high level of communication among group is kept.

Erlang learning has highest level of impact damage in the list since Funky Torrent is being built by Erlang programming language. On the other hand, thanks to lectures to learn Erlang is easier which reduces likelihood of occurrence. The same situation is applied for Software Architecture learning but less impact.

The Group has to consider of bittorrent protocol training as a potential risk even though all members are deeply concentrated on project, learning new thing about it every new day and sharing what have learned among members.

Both sickness in group or losing a team member has the same and the minimal damage risk point because it is not large-scale project and thereby, anybody in team can take the responsibility of another if it is necessary.

5. References

1. Declan Delaney, 2002, Document Templates For Student Projects In Software Engineering
2. http://en.wikipedia.org/wiki/Software_project_management#Software_development_process (accessed on 27.09.2010)
3. http://en.wikipedia.org/wiki/BitTorrent_%28protocol%29 (accessed on 27.09.2010)
4. TIG029 H10 Project: Software Architecture for Distributed Systems, CoursePM-TIG029.pdf
5. http://en.wikipedia.org/wiki/Erlang_%28programming_language%29 (accessed on 27.09.2010)
6. http://en.wikipedia.org/wiki/Scrum_%28development%29 (accessed on 04.10.2010)
7. TIG029 - Software Architecture for distributed systems.pdf

6. Attachment

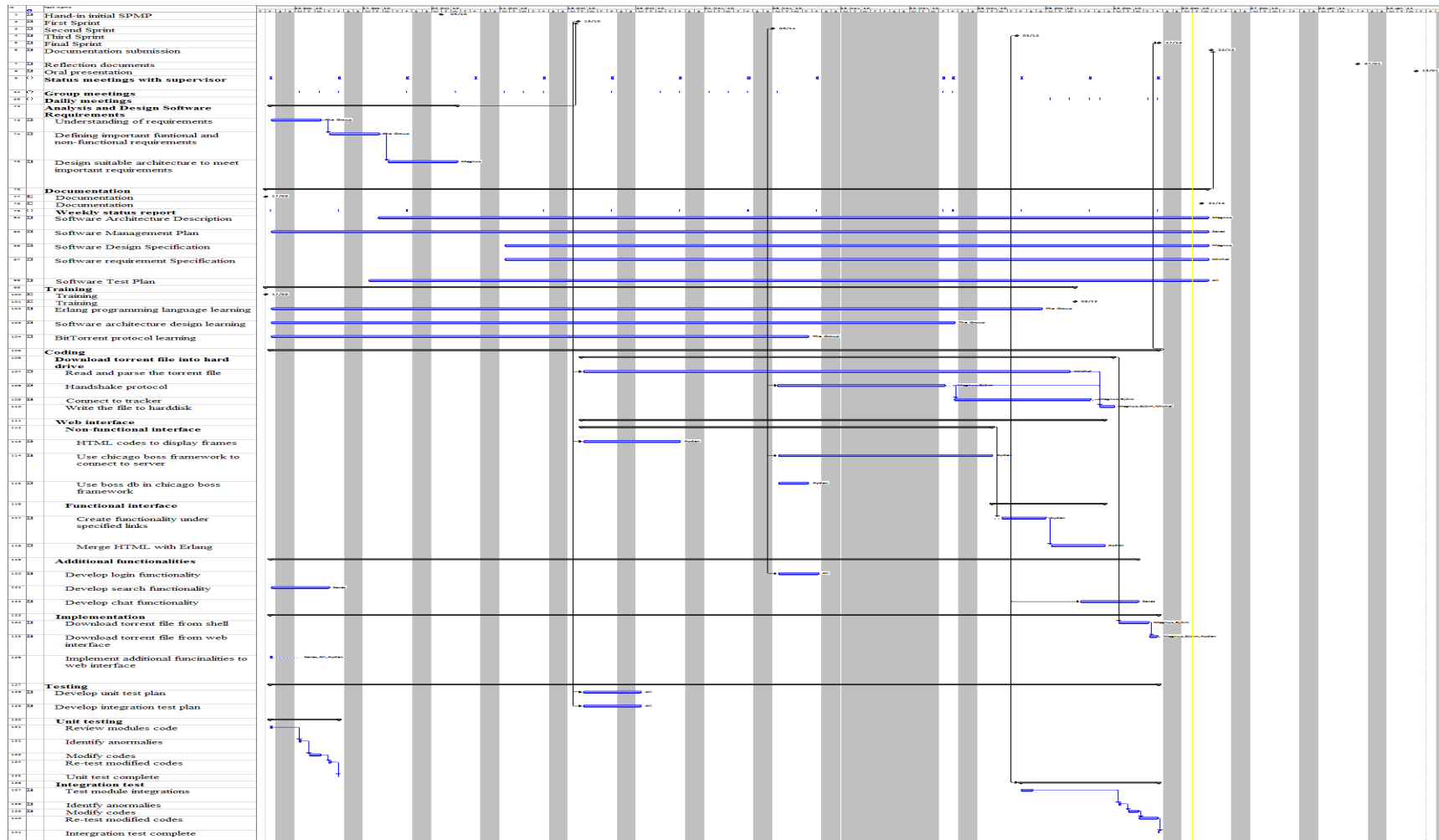


Figure 2. Large-scaled of Gantt chat. (see the The Group Gantt chart in documents