

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Software Patterns Project Plan

Version 1.4

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Revision History

Date	Version	Description	Author
2010-10-01	1.1	Initial Draft	Barua , Chevalier, Borras, Paden
2010-10-28	1.2	Minor changes	Chevalier
2010-11-02	1.3	Revision of chapter 1,2,3,6	Barua
2010-11-29	1.4	Revision of chapter 7,8,9	Barua

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Table of Contents

1. Introduction.....	4
1.1 Purpose of this document.....	4
1.2 Intended Audience.....	4
1.3 Scope.....	4
1.4 Definitions and acronyms.....	4
1.4.1 Definitions.....	4
1.4.2 Acronyms and abbreviations.....	5
1.5 References.....	5
2. Organization.....	5
2.1 Project management.....	5
2.2 Project group.....	5
2.3 Steering group.....	5
2.4 Customer.....	5
2.5 Others.....	5
3. Assumptions and constraints.....	5
3.1 Technological.....	5
3.2 Environmental.....	5
3.3 Interpersonal.....	6
3.4 Work distribution.....	6
3.5 Casual relationships.....	6
3.6 Time.....	6
4. Deliverables.....	6
4.1.1 Remarks.....	6
5. Inputs.....	6
5.1.1 Remarks.....	6
6. Project risks.....	6
7. Communication.....	8
8. Configuration management.....	8
9. Project plan.....	9
9.1 Time schedule.....	9
9.1.1 Remarks.....	9
9.2 Activity plan.....	9
9.3 Financial Plan.....	10
9.3.1 Remarks.....	10

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

1. Introduction

Software Patterns are descriptions of accepted reusable solutions of problems that occur over and over in software design. The project group PG POSE in UPB developing a tool to formally specify a Software patterns, apply, view applied patterns and recognize applied patterns in unknown code. This tool will effectively organized patterns that are specified by this tool and for this an additional tool is required. The tool that we develop in DSD project helps to describe, manage, categorize and compares and show relationship of patterns.

1.1 Purpose of this document

The purpose of this document is show the intent of the project Software Patterns and its complete schedule. This document will also provide the information on team members' roles and their responsibility and work distribution between team members. This also gives an idea of what tools and technology will be used in the project to the team members and other project related stockholders and the deliverables that will have to submit during project and also assumptions on project risks.

This project is being conduct jointly by MDH, FER and UPB as a part of distribute software development course.

1.2 Intended Audience

The intended Audience is teachers, classmates and all those interested and authorized people in our project like investigators, customers, developers, all stakeholders. From this documents all team members will know their roles and responsibilities, supervisors and course teachers and stockholders will know the team organization and get overview of the project.

1.3 Scope

The scope of the plan is within the project of software patterns give the entire activity plan, the guidelines to follow in an organized manner until reach the end of the project, assumptions on technology, tools and environment and also project risks.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Software Patterns	Software Pattern is a reusable solution to a commonly occurring problem in software design. It is not a finished design that can be transformed directly into code but it can be used as template to solve a problem that can be used in many different situations.
Iterative development	Iterative development is an approach to building software in which the overall lifecycle is composed of several iterations in sequence.
Rational Unified Processing	The Rational Unified Process (RUP) is an iterative software development process.
Pattern-Catalog	Contains an arbitrary number of patterns in categories
Pattern	Represent a single pattern
Category	Has an arbitrary number of subcategories
Relation	Refers to a source Pattern and a target Pattern
Iterator	Is often applied to data structures like the design pattern composite
Intent	Briefly introduces the pattern, its use, its intent, problem the pattern is addressed to.
Motivation	A scenario that illustrates a design problem and how the class and object structures in the pattern solve the problem
Applicability	the situations in which the pattern can be applied and how such situations can

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

	be recognized
Structure	A graphical representation of the classes in the pattern by typically using a UML diagram
Participants	the classes and/or objects participating in the pattern and their responsibilities
Collaborations	How the participants collaborate to carry out their responsibilities
Consequences	Describes the trade-offs and results of using the pattern
Implementation	Pitfalls, hints, or techniques which should be considered when implementing the pattern
Sample code	Code fragments that illustrate how the pattern could be implemented
Known uses	Examples of the pattern found in real systems
Related patterns	Which design patterns are closely related to this one and what their differences are

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
MDH	Mälardalen University, Västerås, Sweden
FER	Faculty of Electrical Engineering and Computing, Zagreb, Croatia
UPB	University of Paderborn
PG POSE	Project Group Pattern-Oriented Software Engineering
DSD	Distributed Software Development
IDT	School of innovation, Design and Engineering
GoF	Gang of four
EMF	Eclipse Model Framework
GEF	Graphical Editing Framework for Eclipse
SVN	Subversion revision control software
RUP	Rational Unified Process

1.5 References

1. Specification of Software Patterns
<http://www.fer.hr/rasip/dsd/all> -> project proposals
2. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.
Design Patterns: Elements of Reusable Object-oriented Software.
Addison-Wesley, 1995.
3. <http://www.fer.hr/rasip/dsd/all>

2. Organization

The Software Patterns project is a distributed project among PUB, FER and MDH. The PUB is the PG POSE group and there are six members, FER team is consisted of four members and MDH has three members. Since the project supervisors are from Germany so the project leader is also from PUB and two team leaders from each of FER and MDH. The project group PG POSE will provide a tool to formally specify a design patterns.

2.1 Project management

Project management is the discipline of planning, organizing and managing resources to make a project successful and complete in time. The project manager is the person who will lead the team to achieve the project objectives and team leaders will help him by giving work update time to time.

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Some responsibilities of project leader and team leaders are:

- Manage project in accordance to the project plan
- Serve as liaison to the Steering Committee
- Receive guidance from Steering Group.
- Provide overall project direction.
- Direct/lead team members toward project objectives.
- Handle problem resolution.

The project management team is as follows:

- Supervisor - Dietrich Travkin
- Supervisor - Markus von Detten
- Project Leader - Jan Schmalor
- Team Leader - Shaibal Barua
- Team Leader - Marko Vitas

2.2 Project group

Name	Initials	Responsibility (roles)
Jan Schmalor	JS	Project Leader/Project specification/ Quality assurance/ Integration
Shaibal Barua	SB	Team Leader, Development, Testing
Marko Vitas	MV	Team Leader, Development
Andre Backofen	AB	Project specification/Quality assurance/Integration
Adnan Biser	ABI	Project specification/Quality assurance/Integration
Dennis Nobel	DN	Project specification/Quality assurance/Integration
Joachim		
MarieChristin Platenius	MCP	Project specification/Quality assurance/Integration
Antonio Moreno Borrás	AMB	Development, Backup and Maintenance
Joanne Chevalier	JC	Development, GUI-Design
Jasenko Ramljak	JR	Development, Requirements specification, SVN
Ivica Pađen	IP	Development, Testing
Stipe Grbić	SG	Development, Backup and Maintenance , Testing

2.3 Steering group

Supervisor - Dietrich Travkin
Supervisor - Markus von Detten
Project Leader - Jan Schmalor

2.4 Customer

PG POSE project-University of Paderborn

2.5 Others

Prof. Ivica Crnković (MDH), Prof.dr.sc. Mario Žagar (FER)

3. Assumptions and constraints

3.1 Technological

As this project is actually about developing an Eclipse plugin, the programming language we are going to use is Java.

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Besides, for reports and presentation we chose to use Open Office, and also LaTeX for documentation papers.

3.2 Environmental

The software development will be made using Eclipse and its frameworks :

- Eclipse IDE
- Eclipse Modelling Framework
- JUnit testing tools
- EclEMMA – test coverage

Many programming tools exists for Eclipse and will be really helpful for developing this plugin. This is one of the advantages of using Eclipse IDE.

Besides, Eclipse is a free and multi-platform IDE and the final product will be a plugin that every programmer will be able to use, no matter which operating he uses.

3.3 Interpersonal

Communication is the most important part of this distributed software development project. Each member of the team should keep in mind that it would be very dangerous to neglect this aspect.

The Paderborn members must be frequently informed about the project evolution and the developing decisions so that they can detect problems soon enough and guide the developers telling them what is going wrong.

In order to do that, weekly meetings are organized on Thursdays with the whole team and using Adobe Connect. Also there is a Wiki for the software patterns project where all news and information will be published.

Also there is a project management tool called RedMine installed.

Besides, between those meetings, developers can ask questions to the Project Leader in Paderborn to have more details on what they don't understand, and they are also encouraged to often talk to each others.

Good interpersonal communication essential to complete this project successfully.

3.4 Work distribution

Each member chose the field in which he feels the more comfortable or interested in.

Paderborn members do not take part in the implementation work. Except the meta-model implementation they provide and that is the basis of the Software Patterns plugin. All the other members are developers and must make the documentation related to their work. From each team of FER and MDH there is one person will work on testing.

3.5 Causal relationships

To work in a distributed environment communication is very important and for this it is not enough to have formal relationships among team members. They are encouraged to strengthen their relationship by sharing their ideas, problems etc during the project.

3.6 Time

Each member must read carefully the deliverables table and the activity plan and keep in mind the deadlines.

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Time is one of the most important things to be aware of during this project. Delays can not be accepted and must be avoided.

So it is essential for the team members to be warned when someone is going to be unavailable during a certain period (exams, trip, illness, ...etc.)

4. Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Customer, Supervisors	Project plan document	39	39	0	39	
Customer, Supervisors	Requirements definition document	39	39	0	39	
Customer, Supervisors	Design description document	40	40			
Customer, Supervisors	Acceptance test plan	49	49			
Customer, Supervisors	Test report	2	2			
Customer, Supervisors	Final project report	2	2			
Customer, Supervisors	Final product	2	2			

4.1.1 Remarks

Remark Id	Description
1	Deliverables "Summary week report" and "Minutes of meeting" will be delivered on weekly basis
2	Revisions of existing documents and yet undefined technical documents are going to be delivered during the project

5. Inputs

From	Required item	Planned week	Promised week	Late +/-	Delivered week	Rem
Customer	Project proposal	36	36	0	36	
Customer	Meta-model document	38	38	0	38	
Customer	Use cases document	38	38	1	39	
Customer	Project plan example	38	38			

5.1.1 Remarks

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

Remark Id	Description
1	Project plan example refers to Microsoft Office Project file created by the customers on their other project.

6. Project risks

Possibility	Risk	Preventive action
20%	Failure to respect the deadlines	Distributed and parallel work. Approach while performing tasks must be thoughtful and diligent.
30%	Communication problems	Team members should try to be as most available as it is possible. Meetings should be scheduled frequently. Most important points of audio-visual communication should be written.
20%	Low knowledge	Exchange of knowledge between team members. Team members should prepare for specific task in advance.
20%	Low experience	Exchange of experience between team members.
10%	Unclear project requirements	Documentation should be written correctly and in detail.

7. Communication

Due to the nature of this project there are many ways of communication :

- For video-conference meetings there is virtual room set up every week. The room is accessed through Adobe Connect Pro client.
- There are also project team wiki pages on which it is possible to put informations and news regarding project progress.
- For instant messaging, team members use Skype and Windows Live Messenger. Instant messaging and emails are the most frequent ways of communicating.
- Until there was fixed date for team meetings Doodle tool was used to determine the schedule of the meetings.
- Team members that are currently in the same country communicate via cell phones also.

8. Configuration management

Official DSD Subversion server is used for version controlling of projects code files and documentation. One member of a team is responsible for repository management and maintenance. Other code files and documents exchanges will be made through emails and instant messengers. Certain versions of project solution and documents will be uploaded on projects wiki pages.

Subversion repository is accessible through: `svn://lapis.rasip.fer.hr/svn/dsd10/SoftwarePatterns`

Committing policies can be found in the repository.

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

9. Project plan

9.1 Time schedule

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr.	Rem.
			Plan	Forecast		Actual		
				Week	+/-			
M001	Project Vision presentation	IP	38	38	0	38	0	
M002	Project plan preparation and presentation	SB, AMB, JC	39	39	0	39	0	
M003	Requirements Definition document	MV, JR, IP,SG	39	40	1	40	1	
M004	Design Description document	MV, JR, IP, SG, SB, AMB, JC	41	41	0	41	0	
M005	Alpha Prototype	Team	43	43	0	43	0	
M006	Beta Prototype	Team	45	46	1	46	1	
M007	Acceptance Test Plan	IP, SB, AMB, JC	49	49	0	49	0	
M008	Release candidate	Team	49	49	0	49	0	
M009	Final Project Report document, presentation	Team	2	2	0	1 and 2	0	1

9.1.1 Remarks

Remark Id	Description
1	Some documents will complete before final documentation

9.2 Activity plan

Activity	Phase															
	Inception		Elaboration				Construction				Transition					
	W 39	W 40	W 41	W 42	W 43	W 44	W 45	W 46	W 47	W 48	W 49	W 50	W 51	W 52	W 01	W 02
Project preparation																
Requirements analysis & definition																
Design Description Document																
Implementation																
Testing and bug fixing																
Integration																
Documentation																
Final Project Delivery																
Final Product , report and presentation																

Software Patterns	Version: 1.4
Project Plan	Date: 2010-11-29

9.3 Financial Plan

Activity	Volume (days)	Cost	Rem.
Project Preparation	7	0	
Requirement Analysis and Definition	10	0	
Project Design	15	0	
Implementation	45	0	
Testing	20	0	
Documentation	30	0	
Final Delivery and Presentation	15	0	

Planned effort (man-days)	Man-day cost	Planned project cost (100%)
157	0	0