

SPMP Version 1.0

Contents

Introduction.....	2
Project Overview.....	2
Project Deliverables.....	3
Evolution of the Software Management Plan.....	3
Reference Materials.....	3
Definitions and Acronyms.....	4
Project Organization.....	4
Process Model.....	4
Organizational Structure.....	5
Organizational Boundaries and Interfaces.....	5
Project Responsibilities.....	6
Managerial Process.....	6
Management Objectives and Priorities.....	6
Assumptions, Dependencies, and Constraints.....	6
Risk Management.....	7
Monitoring and Controlling Mechanisms.....	8
Technical Process.....	8
Methods, Tools, and Techniques.....	8
Software Documentation.....	9
Project Support Functions.....	9
Work Packages.....	10
Work Packages.....	10
Dependencies.....	10
Resource Requirements.....	10
Schedule.....	11
Additional Components.....	12

Introduction

Project Overview

Objectives

The object of the WebShell project is to produce a product that satisfies all of our customers requirements and deliver it on time.

Product to be delivered

The product to be delivered is called WebShell. It combines the functionality of a WWW browser, like Netscape or Internet Explorer, with the features of a shell, like a UNIX shell or a Telnet terminal.

Activities

The activities involved with the WebShell include:

- Detailed Design
- Implementation
- Testing
- Integration
- Maintenance

Work Products

The work products produced by the above activities include:

- Detailed design document.
- Source code.
- SQA plan.
- Test cases.
- Test results.
- Known bug list.
- Compiled and deliverable machine code.

Milestones

1. SPMP complete.
2. Detailed design complete.
3. Implementation complete.
4. Alpha complete.
5. Beta complete.
6. Pre-release version complete.
7. Delivery to customer complete.

Required Resources

- 3 Experienced software engineers.
- 3 UNIX work-stations.
- 3 Installations of Java JDK 1.2.
- 1 Solaris compatible printer.

Master Schedule

03 / 02 / 1999	SPMP
03 / 04 / 1999	Detailed Design
03 / 08 / 1999	Alpha
03 / 14 / 1999	Beta

03 / 17 / 1999 Pre-release version
3 / 18 / 1999 Delivery to customer

Master Budget

Students work for free (almost):

Development costs	\$0.00
Anti-Depressants	\$200.00
Alcohol	\$500.00
Caffeine	\$100.00
Entertainment Expenses	\$1,000.00
Total (cash please)	\$1,300.00

Project Deliverables

The WebShell product 03 / 18 / 1999

A JAR package named webshell SlothWare.jar containing all of the classes necessary to run the WebShell product. This JAR package will contain the core class Sloth.

Evolution of the Software Management Plan

If a member of the engineering teams feels that the SPMP needs to be revised he/she must call a meeting with all the members of the engineering team. At this meeting the validity of the revision will be discussed and decided upon. If the revision is approved the engineer who called the meeting shall revise the SPMP. After doing so another meeting will be called with all the members of the engineering team. The correctness of the revision will be discussed and decided upon. If the revision is approved the new SPMP will be given a new incremented version number. If the revision is not approved the engineer may rewrite the revision and call another correctness approval meeting.

Reference Materials

WebShell: Project Description (Versions 1.0 – 1.2)
WebShell: Object Oriented Analysis (Versions 1.0 – 1.1)
WebShell: Specification (Versions 1.0 – 1.1)

Definitions and Acronyms

MRV = Most Recently Visited Queue
HTTP = Hyper Text Transfer Protocol
URL = Universal Resource Locator
CRC = Class Responsibilities Collaborators
I/O = Input / Output
OOA = Object Oriented Analysis
SPMP = Software Project Management Plan
GUI = Graphical User Interface
SQA = Software Quality Assurance
JAR = Java ARchive
JDK = Java Development Kit
FoNoACM = Founder of the Nation of the ACM
ACM = Association of Computing Machinery
IEEE = Institute Electrical and Electronics Engineers
CSIL = Computer Science Instructional Laboratory

Project Organization

Process Model

Activities

- Product development
The project will be divided into three areas. Mihai will be responsible for the control components, Mike will be responsible for the command line components, and Don will be responsible for the GUI components and the executable class.
- Product testing
In the first round of testing, each engineer will be responsible for testing his respective area. The next round of testing will be after product integration. This round will be testing the entire product and all engineers will participate in product testing.
- Product integration
Mihai will be the lead integration engineer since all components must be controlled by his components. Mike and Don will assist in the integration.

Functions

- Software Quality Assurance: since our group is extremely small, SQA will be done as a group managed by Mike.
- Code reviewer: Mihai will lead the code review due to his greater familiarity with Java. Mike and Don will participate as well.

Organizational Structure

The group will be split in sub-groups, working independently on project modules as described in product development. Each sub-group is self-managed. The management of different aspects of the product are defined in Organizational Boundaries and Interfaces below.

Organizational Boundaries and Interfaces

Due to the size of our software group, management will be divided in the following manner: The Product Manager is Mihai. Mike and Don will follow Mihai's direction of the product. The Documentation Manager is Don. All documentation created by Mihai and/or Mike will be reviewed by Don. The SQA Manager is Mike. All organizational decisions, that involve SQA, will be at Mike's direction. The software engineers are Mihai, Mike and Don. The areas of software development responsibility will be determined by the Product Manager with input by the Documentation and SQA Managers. Interfacing with the customer may be done by any software engineer, but a report of the meeting will be shared with the engineers not present.

Project Responsibilities

Product Manager	Mihai Christodorescu, FoNoACM
Documentation Manager	Donald Paul Adams, Monsignor
SQA Manager	Michael Robert Weaver, Esquire
Development	Mihai Christodorescu, FoNoACM Michael Robert Weaver, Esquire Donald Paul Adams, Monsignor
Unit Testing	Mihai Christodorescu, FoNoACM Michael Robert Weaver, Esquire Donald Paul Adams, Monsignor
Integration Engineer	Mihai Christodorescu, FoNoACM
Technical Writer	Donald Paul Adams, Monsignor
Global Visionary	Michael Robert Weaver, Esquire
Entertainment Board of Directors	Mihai Christodorescu, FoNoACM Michael Robert Weaver, Esquire Donald Paul Adams, Monsignor
Computational Historian	Donald Paul Adams, Monsignor
Guru of Paradigms	Mihai Christodorescu, FoNoACM

Managerial Process

Management Objectives and Priorities

Philosophy

Producing great software while maintaining the utmost level of harmony between coworkers and with the client.

Goals

To produce the best possible product within the given constraints.

Priorities

1. Customer satisfaction
2. Employee education
3. Employee happiness

Assumptions, Dependencies, and Constraints

Assumptions

The product will be run on a UNIX or Windows95/98/NT platform. There will be no support for "sound" files.

Dependencies

The product will be developed with JDK 1.2. The browser supported by the product is Netscape due to CSIL limitations.

Constraints

A successfully tested version must be shippable by March 18th, 1999.

Design has to be kept open for multiple platforms, but implementation only for UNIX and Windows95/98/NT using Java and Swing.

A porting document needs to name all platform dependent classes and methods. The document will include only those will have to be changed for a different platform.

Risk Management

Risk Factors

- Missing deadlines
The Product Manager will have overall responsibility for seeing that all deadlines will be met. If a deadline will not be met, the Product Manager will, after discussion with the engineers involved, will attempt to correct the situation and/or handle resulting communications with the client.
- Not meeting clients requirements
The SQA Manager will have overall responsibility for assurance of meeting the requirements. If the SQA manager feels that the requirements are not being met, he will inform the Product Manager. The Product Manager, after consulting the software engineers involved, will attempt to correct the requirements problem.
- Degradation of development quality due to outside pressures (i.e. other classes, real jobs, mental and physical health, etc...)
The Product Manager will have overall responsibility for managing this risk factor. All software engineers are required to inform the Product Manager as soon as possible if development is threaten by outside pressures. The Product Manager will, after discussion with the engineers involved, will attempt to correct the situation and/or handle resulting communications with the client.
- Unsatisfactory communication with the client
The Documentation Manager will have the first responsibility for insuring that communications with the client are satisfactory. If a problem does arise, the Documentation Manager will inform the Product Manager and the Product Manager, after consulting the software engineers involved, will attempt to correct the communication problems.

Monitoring and Controlling Mechanisms

E-mail updates

The primary source of communication and documentation of communications will be e-mail. It is the responsibility of all SlothWare employees to inform, by e-mail, all other employees of any communication about the product.

Peer review

The Product Manager will be responsible for conducting review of the software engineers when necessary.

Grades on assignments

Grades on assignments will be used as yardsticks to gauge client satisfaction upon meeting a milestone.

Communication with the client

Communication the the client will another source of monitoring the client's satisfaction.

Technical Process

Methods, Tools, and Techniques

Tools

- Development System:
 - Windows 9x/NT
 - Dual-Intel Pentium Xeon 450 MHz
 - 128 MB RAM
 - Java Development Kit 1.2 (Java 2)
- Target System:
 - Solaris 2.6
 - Intel Pentium 166 MHz
 - 48 MB RAM
 - Java 2
- Development Environments:
 - Visual J++ 6.0
 - Notepad
 - XEmacs 20.12
 - JDK 1.2
- Programming Language:
 - Java
- CASE Tools:
 - Nada

Methods

- Documentation:
 - JavaDoc

- Programming:
 - Object–Oriented
 - Booch OOA
 - Design Patterns
 - Java Code Conventions by JavaSoft

(<http://www.javasoft.com/docs/codeconv/html/CodeConventionsTOC.doc.html>)

Techniques

- Testing:
 - Code Coverage
 - Black Box

Software Documentation

Baseline

- IEEE 1058.1 SPMP (omitting the Staffing Plan and the Budget and Resource Allocation)
- JavaSoft JavaDoc Guidelines
(<http://www.javasoft.com/products/jdk/javadoc/writingdoccomments.html> and
<http://www.javasoft.com/docs/codeconv/html/CodeConventions.doc4.html#16838>)

Milestones

The first milestone is the completion of the SPMP. Then the completion of the detailed design. After that at least three reviews of the JavaDoc output, the SPMP, and the detailed design will take place. These reviews will take place after each of the following project milestones are met: Alpha, Beta, and Pre–Release.

Project Support Functions

Configuration Management

Configuration Management will be done by the Product Manager. He will inform all software engineers when a change needs to be made. All software engineers are responsible for informing the Product Manager when they are requesting a updated files to be added to the Product.

Quality Assurance

Quality Assurance will be managed by the SQA Manager as described in Organizational Boundaries and Interfaces.

Testing Plans

Testing plans will be conducted as described in Product testing.

Work Packages

Work Packages

GUI

- Tree display functional module
- File display functional module

Console

- Command interpreter
- History module
- Alias table
- Environment Variable table

Control

- Document tree module
- MRV module
- File system module

Web

- HTML parser
- HTTP connect module

Dependencies

The GUI and Console modules depend on the Control module. The Command interpreter depends on the History module, the Alias table and the Environment variable table. The Control module depends on the Web module, and the Document tree module.

Resource Requirements

CSIL at UCSB

Subject to posted hours.

Hardware

Will use PCs, printers and other devices provided in CSIL.

Software

Will use software provided in CSIL.

Software Engineers

Will use engineers that are employees of SlothWare.

Management

Will use engineers that are employees of SlothWare.

Human resources usage chart

(please see next page)

<i>Time and Task</i>		<i>Utilization</i>		
		Mihai	Mike	Don
02/28/1999 – 03/02/1999				
	SPMP	50.00%	50.00%	90.00%
	SQA	50.00%	50.00%	10.00%
03/02/1999 – 03/04/1999				
	Detailed Design	50.00%	50.00%	90.00%
	SQA	50.00%	50.00%	10.00%
03/04/1999 – 03/17/1999				
	Implementation	70.00%	25.00%	50.00%
	Unit Testing	25.00%	15.00%	20.00%
	SQA	5.00%	60.00%	30.00%
03/18/1999				
	Delivery	100.00%	100.00%	100.00%
03/19/1999 – 03/24/1999				
	Maintenance	100.00%	100.00%	100.00%

Schedule

Activity	Deadline
Tree display	
Command interpreter	3/2/99
MRV module	
File display	
History module, alias table	3/4/99
File system module	

Activity	Deadline
GUI module integration	
Environment Variable table	3/6/99
Document tree module	
HTTP connect module	
Alpha version demo	3/8/99
Web module integration	3/10/99
Control integration	3/11/99
Product integration	3/12/99
Beta version demo	3/14/99
Product testing and fixing	
Pre-release	3/17/99
Delivery	3/18/99
Maintenance	3/24/99

Additional Components

De Nada component, as specified above.