CS 673
Assignment #3
Software Configuration Management Plan (SCMP)

Team Snow Crash
Dale Earnest
Jeff Dunn
Dong Luo
Mike McWilliams

Table of Contents

Revision History

Name	Date	Reason For Changes	Version(s)
Dale	29 Sept	Initial version	0.1
Dale	03 Oct	Additional Revisions	0.2
Dale	05 Oct	Post Mortem revisions, time added	0.3
Dale	06 Oct	License	0.4
Dale	06 Oct	Final version	1.0

1.Introduction

This Software Configuration Management Plan (SCMP) describes how the artifacts for the evolutionary simulation project is to be managed.

1.1Acronyms

CI: configuration item – an item tracked by the configuration system

CL: Configuration Leader – manages the configuration plan and code repository.

CM: configuration management – the process of maintaining the relevant versions of the project.

I/O: Input/Output and file handling system.

SCMP: the Software Configuration Management Plan (this document).

UI: User Interface of the application.

1.2Terms

Approved CI's: CI's signed off by project management.

Artifact: A final or interim product of the project e.g., a document, source code, object code, test result

Commits: A item or artifact stored in the code repository.

2.SCM Management

2.1Organization

One of the team members, Dale Earnest, is designated the configuration leader. All others are designated engineers.

The backup configuration leader is Mike McWilliams; should there be a case where the CL is unable to respond to a question within two days, the backup CL can take on the duties of the CL.

2.2SCM Responsibilities

2.2.1Configuration Leader

It is the responsibility of the configuration leader to:

- Set up, organize, and manage the SCMP.
- Set up, organize, and manage the code repository.
- Maintain, back up, and assist in resolving issues with the code repository.
- Create a backup plan and repository should the current tool be unsupported.
- Resolve conflicts between engineers regarding "commits."

Additional responsibilities of the configuration leader, are stated in sections 3.3, 3.4, 3.5, and 3.6.

2.2.2Engineers

It is the responsibility of each engineer to abide by the CM rules which the configuration leader publishes.

Engineers are further divided into three areas; engineers may commit code within their area without prior review by the CL. An engineer wishing to commit code into a different area must get prior approval by the engineers working in the other area and the CL.

The three areas are:

- UI: Jeff Dunn
- Application Logic: Mike McWilliams, Dale Earnest
- I/O: Dong Luo

Additional responsibilities of the engineers are stated in section 3 below.

2.3Applicable Policies, Directives, and Procedures

1. Every other week in the outside class meetings policies and procedures will be reviewed to ensure that said policies are propelling development and not hindering development.
2. Improvements to these guidelines will be posted within two days of team agreement.
3. All current and previous releases of this document are retained.
4. The Evolutionary Simulation project will use the Google Code repository.
5. The Google Code repository will be accessed through the Mercurial Eclipse plug-in (version 1.6.0).
6. Authentication is managed according to the Google Code repository password system.
7. All commits are to include release notes sent to the entire team to notify them of changes.
8. The code is to be licensed under the Artistic GPL.

Code formatting is as follows:

1. Curly braces are the same line.
2. Always use curly braces to enclose logical statements.
3. Camel case variable declarations.
4. Keep classes, methods, and variables to the smallest scope.
5. Declare class variables as private.
6. Use inline and block comments for code whose purpose or function is not obvious.
7. If something is public, use java docs.

3.SCM Activities

This section identifies all functions and tasks required to manage the configuration of the software system. Both technical and managerial SCM activities shall be identified.

SCM activities are generally grouped into four functions: configuration identification, configuration control (change control), configuration status accounting (reporting) and configuration audits and reviews.

3.1Configuration Identification

3.1.1Identifying Configuration Items

The CL shall be responsible for identifying all non-Java source CI's. Engineers wishing to propose non-Java source CI's shall secure his or her agreement through supporting documentation. E-mail is an acceptable form for this.

All other Java source CI's are subject to weekly review by the CL to ensure conformity with team policies.

Configuration items are:

- Java source code
- project requirements
- project documents
- 3rd party code libraries
- image files.

3.1.2Naming configuration items

The CL shall be responsible for ensuring all CI's conform to naming conventions.

File conventions are:

- Root directory: teamsnowcrash
- Source directory: src
- General document directory: doc
- UML directory: doc/UML
- Homework directory: doc/hw
- 3rd party library directory: lib
- Java files: XX.java (where “XX” is the name of the class)
- Image files: image (PNG format)

Text documents will be in ODT format. UML will be in DIA format.

3.1.3Acquiring configuration items

All CI's are acquired through use of the Mercurial Eclipse plug-in.

Any troubles accessing the code repository should be reported to the CL.

3.2Configuration control

3.2.1Requesting changes

Engineers may commit code within their area without prior review by the CL. An engineer wishing to commit code into a different area must get prior approval by the engineers working in the other area and the CL.

3.2.2Evaluating changes

Each weekly meeting shall have a mini code review where the week's changes are reviewed. It is expected that each team member be familiar with the state of the previous day's code.

3.2.3Approving or disapproving changes

During the weekly evaluation, conflicting changes are discussed. Any changes not discussed are considered approved. Changes that require discussion will be either approved or disapproved that meeting.

3.2.4 Implementing changes

Baseline version releases are set for these cases:

- 0.1: Configuration screen
- 0.2: Run Simulation
- 0.3: Results screen
- 1.0: Final build

Once baseline releases are set, changes to those releases will be reviewed first before integrating the changes into the mainline as team members are dependent on these releases being stable and fixed.

3.3Configuration status accounting

Current release notes will be published on the Google Code site as well as the team wiki.

3.4Configuration audits and reviews

The CL will present the current status of the project in the weekly meetings.

3.5Interface control

N/A.

3.6Subcontractor/vendor control

N/A.

4.SCM Schedules

	October				November					December	
	4	11	18	25	1	8	15	22	29	6	8
CM Setup	x										
CM Review	x	x	x	x	x	x	x	x	x		
Baselines			0.1			0.2			0.3	1.0	

5.SCM Resources

Configuration leader will require an estimated average of three hours a week to maintain the system configuration. We have chosen not to call out separately the time spent by the other team members on configuration management.

6.SCM plan maintenance

Due to the importance of a stable SCM plan, all changes to this document must be approved by the entire CM team.

7.Hours

7.1Total Hours

Deliverable	Estimated Hours	Actual Hours
STCD	30	25
SPD	40	38
SCMP	40	57
SPMP	40	
SRS	60	
SDD	40	
Group Project Presentation	30	
Individual Project Report	10	

7.2Hours Broken Down

Item	Dale Earnest	Mike McWilliams	Jeff Dunn	Dong Luo	Total Hrs.
Meetings	6	6	6	6	24
Communications Plan	0	0	0	0	0
Application Investigation	2	3	4	2	11
Prototype (coding)	2	1	0	0	3
Prototype (drawings)	1	1	8	0	10
Project Proposal	1	0	0	0	1
Document Editing	4	0	0	0	4
Document Review	1	1	1	1	4
Total Hours	17	12	19	9	57

8.Post Mortem

Before reviewing the document, we estimated that we would find 2 defects per document page. Defects would consist of:

- Statements that were factually incorrect.

- Factual items missing from the plan.

After the review, we found 9 defects. This makes for an estimate of 1 defect per document page, half what we expected.