Improving Visualization Knowledge (VisKo)

Software Configuration Management Plan

Version 1.1

2/12/14

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | February 4, 2014 |
| Current Release: | V1.1 |
| Indicator of Last Page in Document: |  |
| Date of Last Review: | February 3, 2014 |
| Date of Next Review: | February 4. 2014 |
| Target Date for Next Update: | February 4, 2014 |

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Yoonsik Cheon

Bhanukiran Gurijala

Customer: Dr. Nicholas Del Rio

Software Team Members:

Maria Cortes – V&V Lead

Manuel Dosal – Systems Designer

Mario Flores – Lead Programmer 1

Rebekah Gruver – Lead Programmer 2

Hector Quintana – Systems Architect

Marcela Vazquez – Systems Analyst

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 2/2/2014 | Leaf Development | Version 1.0 of completed SCM compiled by the entire team |
| 1.1 | 2/4/2014 | Leaf Development | Refined and detailed all sections of SCM |
|  |  |  |  |

Table of Contents

Document Control 1

Approval 1

Document Change Control 1

Distribution List 1

Change Summary 1

1. Introduction 3

1.1. References 3

2. Software Configuration Identification 4

2.1. Software Configuration Item Identification 4

2.2. Software Configuration Item Organization 4

2.2.1. Naming Convention 4

2.2.2. Labeling Scheme 4

2.2.3. Management 5

2.2.4. Detailed Directory Structure 6

2.2.5. Back-Up 6

3. Software Configuration Control 7

3.1. Documentation 7

3.2. Configuration Control Board 7

3.3. Procedures 7

4. Software Configuration Auditing 9

5. Appendix A 12

6. Appendix B 13

7. Appendix C 14

# Introduction

The main purpose of the project this semester is to implement the new software requirements to improve the widespread adoption of Visualization Knowledge (VisKo) system, by expanding the current VisKo system features to include pipeline and visualization browsing, ability to issue queries, and ability to set parameters and pipelines in a simplified manner [1]. The Software Configuration Management (SCM) will help the team with quality assurance by tracking and controlling changes in the software. Some of the techniques to control the changes include revision control and the establishment of baselines. Following the SCM will detail the exact change that occurred and the person responsible for it [2].

The intended audience of this document is the guidance team, in order for them to assist the team in defining a process for SCM. At the same time this document is intended for the developer team, as a guide through all the process of implementing and coding of this project.

This document is composed of three main sections:

1. Software configuration identification. This section provides a list of items likely to be changed throughout the life of the project. These items include source code, test cases, design documents and requirements. It also describes the labeling convention and how the team will be collaborating in the updates. It includes a visual description of the convention used to label the updates, as well as how changes will be requested. At the same time, this section describes how the team backup process in order to avoid total or partial loss of documents or source code.
2. Software configuration control. Describes the mechanism to prepare and evaluate the proposal for changes to the configuration items throughout the life cycle. The control board will approve or disapproved all change proposals when a team member fills out a change request form and justify the means for doing it. It also describes the process to be followed to control access to source code and documents in order to avoid unauthorized updates and collisions between versions if more than one team member are working on the same task simultaneously.
3. Software configuration auditing. This section provides detail of the two types of auditing: Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA). Configuration audits provide the status of the project by evaluating the configuration items. In addition, a traceability matrix will be used to ensure that all requirements are being met in the implementation of the project.

## References

[1] Interview Report.

[2] <http://en.wikipedia.org/wiki/Software_configuration_management>

[3] Cheon, Yoonsik,*Software Configuration Management (SCM).* [Microsoft Power Point] El Paso, TX, 2014.

[4] Guidance Team, “Software Configuration Management Plan Template,” El Paso, TX, 2014.

[5][http://www.westfallteam.com/sites/default/files/papers/Software\_Configuration\_Management\_Audits.pdf](https://bl2prd0511.outlook.com/owa/redir.aspx?C=ArAxK95bJEKTGM-diZ-MgpXUoNmDhNAIc22EG2EPMF8B3Ina264x83WnRN5aJf8ZatipKO8QmFk.&URL=http%3a%2f%2fwww.westfallteam.com%2fsites%2fdefault%2ffiles%2fpapers%2fSoftware_Configuration_Management_Audits.pdf" \t "_blank), “scm-audit.pdf”

# Software Configuration Identification

This section provides labels for the baselines and their updates. Provides a list of items likely to be changed throughout the life cycle of the project. It also includes description of the naming or labeling convention to be used and how the team will be collaborating in the updates. A back up method is described that will be used to avoid total or partial loss of documents or source code.

## Software Configuration Item Identification

Configuration Elements – elements that are likely to change throughout the “Improving Visualization Knowledge” process:

1. Documents
2. Source Code
3. Test Suites

## Software Configuration Item Organization

This section describes the labeling scheme for baselines and updates to baselines for each type of item.

### Naming Convention

|  |  |  |
| --- | --- | --- |
| Configuration Elements | Baseline Format | Example |
| Documents | *documentTitle\_leadPosition* | planReport\_Analyst.docx |
| Source Code | *className* | dataFeature.java |
| Test Suite | *testName\_date\_#* | dbtestOne\_Jan30\_1.jtt |
| Audit Report | *auditName\_PCA*  *auditName\_FCA* | auditOne\_PCA.docx |

Figure 1.0 – Elements Naming Convention

### Labeling Scheme

**Documentation**

v3.0

v2.0

v1.0

v2.2

v2.1

v1.2

v1.1

* Baseline: v1.0, v2.0, v3.0, etc.
* Updates: v1.1, v1.2, v1.3, etc.

**Source Code**

v2.0

v1.0

v1.2

v1.1

v1.1.1

v1.1.2

v1.1.3

* Baseline: v1.0.0, v2.0.0, v3.0.0, etc.
* Group Updates: Updates: v1.1.0, v1.2.0, v1.3.0, etc.
* Individual Updates: v1.0.1, v1.0.2, v1.0.3, etc.

**Test Suites**

|  |
| --- |
| **Labeling** |
| testData\_Jan30\_1.jtt |
| testData\_Jan30\_2.jtt |
| testDB\_Jan31\_1.jtt |
| testTwo\_Jan31\_2.jtt |
| testTwo\_Feb1\_1.jtt |

* Labeled by date and numbered in order by that date (as defined in naming convention).

**Audit Report**

* Every Audit Report is labeled with the version number the team is working.

### Management

**Documents**

As mentioned in 2.2.2, Documents will primarily be stored in the appropriate assignment folder that the team creates. The assignment folder will contain each baseline version/subversion of the document in that folder, as well as the final submitted assignment document.

**Program**

There will be only one program folder that contains multiple versions of source code in that folder.

**Test suites**: specified to each version of the code

Each test suite specific to its appropriate source code will be stored with that source code version and only that version. As mention before, the program folder will contain each version of source code, each with its appropriate test case.

**Audit Report:** specified to each version of the code

Each Audit Report will be stored in each update (v1.1..v1.3 etc) in order to guide the team with their duties. Also, each Audit Report will be stored in the folder where the source code in progress is saved.

### Detailed Directory Structure

* Documents
* Source Code
  + v1.0
    - v1.1
      1. Test Suites
      2. Audit Reports
    - v1.2
      1. Test Suites
      2. Audit Reports
  + v2.0
    - v2.1
      1. Test Suits
      2. Audit Reports

### Back-Up

Primary control of back up will be the teams repository on GitHub, however each member is responsible for maintaining each document, source code, test, etc. on his or her personal computers. Each member is also responsible for maintaining their branch on GitHub, then following the process of merging with others.

Google Drive will also be an additional back up form that contains all our documents/files that have been worked on, but will not be modified through there. Since there will always be online it doesn’t require an actual date to back it up.

# Software Configuration Control

This section provides the detailed mechanism for evaluating and approving or disapproving all change proposals to the configuration items throughout the life cycle. The purpose of this section is to identify what mechanism will be used to control access to items in the configuration in order to prevent unauthorized updates and collisions between team members working on the system simultaneously.

## Documentation

Any member of the team will be able to make any changes they want to the section of the document or code they have been assigned until the due date comes; they will have “control” over the document until the due date. In order to perform any change a group member who has no control will be required to fill out a form and sent it to the control board for approval, the form is shown in Appendix A.

## Configuration Control Board

* + The control board will be the group member or group members in charge of the configuration item.
  + For code the control group will be Programmer 1 and Programmer 2.
  + For test suites the control group will be the V&V leader.
  + The approval process will be as follows.
    1. Any group member who wishes to make a change will fill out a change request form. This form will then be emailed to the control board.
    2. The control board will receive the email and review the change request. They will be required to make a decision in less than 12 hours and let the requester know their decision.
    3. If the change request was approved then the requester will need to complete the changes by the anticipated completion date specified in the request form.
    4. After the changes are done, the requester sends these changes to the control group in order to review and if approved by the control group, they will upload the new version.
  + Unauthorized approvals include changing an item before the control group approves and after a team agreement.
  + In order to authorize any change in the items, it has to pass the Software Configuration Auditing guidelines.
  + The configuration manager group will supervise that the guidelines in this document are upheld. Also, this group will support the product development activity by providing appropriate workspace and resources to build and test the project. The configuration manager group will consist of the System Analyst and the System Architect.

## Procedures

* + The approval process for any new baseline will depend upon guidance team approval. Therefore no document can be considered a baseline until the guidance team has reviewed it and approved it.
  + We will be using a GitHub repository to store code. Each member will have a separate branch that they will be able to freely edit and commit to. The branches will be reset each time the code changes group version to ensure members work on the correct version of the code. When a team member turns in any code they will be in charge of merging their own code with the main branch. If a team member performs an unauthorized merge the code will be reset to the previous version. However, the control group and the V&V will be in charge of testing after changes are made.
  + Any other configuration item will only be directly accessible by the control board. Team members will have to work on their part locally and submit it to the control board through email when done. The control board will then merge this work with the current version of the configuration item. Any work submitted will need to follow the configuration item format, which we will receive from the guidance team. The configuration items will be stored on a Google drive folder in order to have version control.
  + After work has been submitted and merged the control board will need to follow the following guidelines.
    1. If the document is less than 10 pages the control group will then email it to everyone for review.
    2. If it is greater or equal than 10 pages then the control group will be able to submit if the document is acceptable.
    3. If the control group finds an error in one part of the document or code, they have 12 hours to inform to the responsible person. The responsible person will need to check their email through the 12 hours in order to check if any errors were found. Then, he or she has to resubmit a corrected version of the item in the next 12 hours unless specified otherwise. Therefore, the initial submission of each member has to be done one day before the final document is submitted.
       1. An error is only expected to be any item that does not comply with the baseline documents.

# Software Configuration Auditing

This section explains the two types of auditing that will be taking place.  The importance of auditing is that it provides a mechanism for determining the degree to which the current state of the system is consistent with the latest baseline and documentation [3]. At the same time configuration audits provide greater visibility into the status of a project by evaluating the status of the configuration items. Finally, they also determine the traceability from requirements, and change requests to the implementation by investigating the baselines and changes to the baselines [4][5].

The two types of auditing are defined as follows:

* **Functional Configuration Audit** (FCA), which verifies that a configuration item is in accordance with its software requirements.
* **Physical Configuration Audit** (PCA), which ensures that items in the baseline are of the right version.

Each type of auditing procedure will be explained in greater detail in the tables below.

**Functional Configuration Audit (FCA)**

A FCA is performed in a group of three people assigned by the lead of the project, which includes the V & V Lead. Typically the process is performed no more than 1 day after a new configuration item has been identified and named.  The FCA team will look over the configuration item and answer the questions in [Table 1](#table1) along with any other question they feel should be addressed in evaluating the item. The FCA members will list major issues, and inconsistencies in the Audit Report included in Appendix B. At the same time the members may write comments, questions and concerns regarding the issues found on the configuration items. As for small issues found, the members will solve the issues immediately as they go through the checklist on [Table 1](#table1).

**Table 1** Functional Configuration Audit

|  |  |
| --- | --- |
| **Checklist Item** | **Suggestions for addressing checklist item** |
| List all the behavioral requirements from the SRS that are met. | * This checklist item specifically asks if there are any requirements from the SRS that **have** been met in the configuration items. |
| List all behavioral requirements from the SRS that are addressed | * This checklist item specifically asks if there are any requirements that **can** be met by referencing this configuration item. |
| List all the non-behavioral requirements from the SRS that are met. | * This checklist item specifically asks if there are any non-behavioral requirements in the SRS that **have** been met in the configuration items. |
| List all non-behavioral requirements from the SRS that are addressed | * This checklist item specifically asks if there are any requirements that **can** be met by referencing this configuration item. |
| Have all change requests been resolved? | * For un-resolved change requests, the change request may simply need to be closed, or it may need further testing, or deferred to a later baseline. |
| Is the Configuration Item complete? | * Does the item explain everything it is trying to explain, or does it need to be used with another configuration item.  If so, list the configuration items in the Audit Report. * Was there anything you felt was missing from the report? If so lost anything missing from the report, in the Audit Report. * Were there any inconsistencies regarding the software requirements? If so list inconsistencies in the Audit Report. |
| List any questions or concerns. | * List any questions or concerns you may have of the item or the requirements in the Audit Report. * FCA team will identify and list all issues for which action is required in the Audit Report. |
| Were all questions and concerns addressed | * Was there a follow up for the audit? * Did the team members address the issues found in the FCA process? * Team members must resolve all issues found in the FCA process. |

Additionally, in order to ensure all requirements have been met in the implementation, a traceability matrix will be used, and a copy of the traceability matrix will be added as an appendix to the SRS. This method will allow the teams ensure that all requirements are met in the implementation. As well as allow us to review requirements that still need to be implemented.

The traceability matrix will be updated a week after new features have been added to the implementation. The V&V Lead will be in charge of managing these updates. The updates will be completed in the traceability matrix using Excel Spreadsheet from Microsoft as shown in Appendix C. Only the functional requirements that process information are expected to have test cases.

**Physical Configuration Audit (PCA)**

A PCA is performed in a group of two people not assigned to do the FCA; this is because it’s important to have all the team members work on different parts of the audit processes. Doing this will also allow the PCA group to be objective in their assessment of the FCA process. The PCA process typically takes place no more than 2 days after a new configuration item has been identified and named. The PCA team will look over the configuration item and answer the questions in [Table 2](#table2), which states a checklist that should be addressed when conducting a PCA. The PCA members will list major issues, and inconsistencies in the Audit Report included in Appendix B. At the same time the members may write comments, questions and concerns regarding the issues found on the configuration items. As for small issues found, the members will solve the issues immediately as they go through the checklist on [Table 2](#table2).

**Table 2** Physical Configuration Audit

|  |  |
| --- | --- |
| **Checklist item** | **Suggestions for addressing checklist item** |
| All configuration items have been updated to the correct version | * This checklist item specifically asks if each configuration item is named accordingly to the **correct** version as specified in the software configuration identification. * **Example**: the SCM report has been updated to the correct version, after major changes made by the analyst. * List any inconsistencies in the Audit Report. |
| All configuration items are named correctly according to the naming convention | * This checklist item specifically asks if each configuration item is **named correctly** according the naming convention. * **Example**: Documents use the following format: *documentTitle\_leadPosition* * List any inconsistencies in the Audit Report. |
| All configuration items are properly placed in the correct physical location | * This checklist item specifically asks if each configuration item **has been placed** in the correct physical location in the repository. * **Example:** all document configuration items are only placed in the documents, not with source code. * List any inconsistencies in the Audit Report. |
| List any questions or concerns. | * List any questions or concerns you may have of the item or the requirements in the Audit Report. * FCA team will identify and list all issues for which action is required in the Audit Report. |
| Were all questions and concerns addressed | * Was there a follow up for the audit? * Were issues resolved? * Team members must resolve all issues found in the PCA process. |

# Appendix A

**Leaf Development Team**

**Request Change Form**

|  |  |
| --- | --- |
| **Requester Name:** |  |
| **Date:** |  |
|  |  |
| **Code/Document name:** |  |
| **Code/Document Version:** |  |
| **Requested Change:** |  |
| **Reason for Change:** |  |
| **Due Date:** |  |
| **Text/Code file(s):** |  |

|  |  |
| --- | --- |
| **Control Group** |  |
| **Name:** |  |
| **Approved:** | * **Yes** * **No** |
| **Reason:** |  |
| **Signature:** |  |

**Fig. 1 Request Change Form**

# Appendix B

**Audit Report**

**Software Configuration Item Name/Version:**

**Type of Audit Process:**

**Audit Members:**

**Date:**

|  |  |
| --- | --- |
| **List of Inconsistencies**  **Missing Items, Other Issues** | **Comments**  **Questions**  **Concerns** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Appendix C

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Requirement** | **Tested** | **Test Case** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Date:**

**Implementation Date:**

**Comments:**

