PMR Insight Collective Knowledge (PICK)

Software Configuration Management Plan

Version <1.6>

<2/6/2020>

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 1.6 |
| Indicator of Last Page in Document: | $$$ |
| Date of Last Review: |  |
| Date of Next Review: |  |
| Target Date for Next Update: |  |

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. Steve Roach

Jake Lasley

Customer: Mr. Baltazar Santaella

Ms. Herandy Vasquez

Mr. Vincent Fonseca

Ms. Florencia Larsen

Mr. Eric de Nava

Software Team Members: Itzel Rivas

Eder Rodriguez

Jose Gallardo

Gilbert Alvarez

Nusrat Sarmin

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 2/2/2020 | Itzel Rivas | Started the document |
| 1.1 | 2/4/2020 | Itzel Rivas | Added section 1 (introduction) |
| 1.2 | 2/5/2020 | Gilbert Alvarez | Answered: Describe how errors will be documented, how will change requests be made. |
| 1.3 | 2/5/2020 | Jose Gallardo | Answered: What source code control tool will you be using? Where is it hosted? |
| 1.4 | 2/5/2020 | Nusrat Sarmin | Answered: How will you back-up files? |
| 1.5 | 2/5/2020 | Eder Rodriguez | Defined the directory structure that we are currently using. |
| 1.6 | 2/5/2020 | Itzel Rivas | Added questions on how to keep track of the versions of our code- files, and storage of documents and code. Added to section 2.1 |

Table of Contents

[Document Control ii](#_Toc61860422)

[Approval ii](#_Toc61860423)

[Document Change Control ii](#_Toc61860424)

[Distribution List ii](#_Toc61860425)

[Change Summary ii](#_Toc61860426)

[1. Introduction 1](#_Toc61860427)

[1.1. References 1](#_Toc61860428)

[2. Software Configuration Identification 1](#_Toc61860429)

[2.1. Software Configuration Item Identification 1](#_Toc61860430)

[2.2. Software Configuration Item Organization 1](#_Toc61860431)

[3. Software Configuration Control 1](#_Toc61860432)

[3.1. Documentation 1](#_Toc61860433)

[3.2. Configuration Control Board 2](#_Toc61860434)

[3.3. Procedures 2](#_Toc61860435)

[4. Software Configuration Auditing 2](#_Toc61860436)

# Introduction

The purpose of the Software Configuration Management Plan (SCMP) is to give the customer and guidance team a clear and precise description of tracking and controlling changes in the PMR Insight Collective Knowledge (PICK) software. The SCMP divides itself into three further sections: Software Configuration Identification (SCI), Software Configuration Control (SCC), and Software Configuration Auditing (SCA). The SCI is used to identify any items that can be controlled as well as establish their versions and baselines. The SCC is used to identify the procedures that 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. The SCA is used to provide a procedure for determining if the current configuration of the software system mirrors the software system pictured in the baseline and the requirements documentation. The intended audience of the SCMP are our customers and guidance team.

## References

<http://swebokwiki.org/Chapter_6:_Software_Configuration_Management#Software_Configuration_Control>

# Software Configuration Identification

<< This section provides labels for the baselines and their updates.

>>

## Software Configuration Item Identification

<< Identify a complete list of elements that will make up a configuration. A “configuration” is the set of things you need to create and install a working version of your system. These items include source code, design documents, test suites, requirements documents, project plans, project standards, and other documentation such as user guides. It may also include specific versions of COTS products needed to build and run the system. Identify only the items that are to be controlled in a given configuration. >>

List of dependencies that will have to be downloaded to make the program run:

1. Python3
2. PyQt5
3. QGraphViz

## Software Configuration Item Organization

What source code control tool will you be using? Where is it hosted?

The source code control tool we will be using is Git. It will be hosted on our team repository located on GitHub.

Define the directory structure that you are using or will be using to manage documents and program-code files.

To maintain an understandable directory structure for our code, we will follow, to some point, the directory structure of Model-View-Controller Architecture. We maintain our view objects in one folder, our model objects in a different folder, and our Controllers, which in this case is are the Manager objects, in the root directory of our code base.

How will you manage versions of your program-code files? What naming conventions will you use?

To manage the versions of our program-code files we will be version control from GitHub, which will allow us to keep track of our work and helps us to easily explore the changes you we made. For the naming conventions we will try to be as descriptive as possible in the title we give each file, so it will be easier to identify and work with it e.g. graphNode\_V1.

What process will your team use to update group documents and source code?

The process that our team will be using to update documents is through Microsoft Teams, since last semester we have been using this app to store and update documents that we will be working throughout the progress of the class. Also, for the source code we will be using GitHub to store and update our code as we progress throughout the project.

Describe how errors will be documented.

A baseline must be established to determine a current working version of configuration items that the team agrees on. The entire team will identify errors of the different configuration items within the baseline and check for defects, bugs in the code, UI and grammatical issues, and logical issues that prevent the software from preforming the way it should.

How will change requests be made?

Changes, whether they be defects detected by the team or enhancements suggested by the client/instructor/individual team members, are recorded onto a Software Change Request (SCR) document. An impact analysis is performed on the SCR by the team to determine if the new version has the right changes, if any other changes that weren’t part of the SCR appeared on the software, and if the changes proposes are the correct changes. The new change decisions evaluated by a configuration control board, used by the whole team, that would approve or deny change requests.

* Resource: <http://swebokwiki.org/Chapter_6:_Software_Configuration_Management#Software_Configuration_Control>

How will you back-up your files?

To ensure file and folder security, prudence dictates we take a backup. There is a single python file which does the actual backup. We pushed our files to git. So, if mistakenly we delete or loose the file then we can get it from git. If we delete the git repository then we should have the files in our laptop. If someone pushed wrong files to the repository then we can easily get back the previous version of the files in git. That’s how we ensure our files back-up. However, we know another two kinds of backup: Copying files to another disk or folder and Compressing files and folders and saving the zip file.

# Software Configuration Control

<< Provide a detailed mechanism for preparing, 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 mechanisms 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

<< Provide documentation for formally precipitating and defining a proposed change to a software system.

Explain how you will document changes to the configuration. What style will be used? NOTE: In a large software project, this section would include a series of forms or procedures for submitting a change request to a committee for review. A change request form usually contains information related to who is requesting the change, expected start and delivery dates, a description of the change, priority level, business justification for the change and a section to be completed by the development team where an initial assessment is provided on what the impact of the change will be in the system, level of effort needed to complete the change, approval signatures and actual start and delivery dates. This type of documentation assists project managers to maintain and evaluate metrics related to the progress of the project. >>

## Configuration Control Board

<< Provide an organizational body (Configuration Control Board) for formally evaluating and approving or disapproving a proposed change to a software system. Explain who will have access to modifying different parts of the system. How will changes be approved or disapproved, i.e., what factors will be evaluated in order to approve or disapprove a change. Can anyone make changes to any other person’s components? Who will be in charge of distributing changes? NOTE: The Configuration Control Board in your software team is the software team. Describe how V & V will report errors in the code. Describe how changes are approved and distributed. A common approach is to have two people, e.g., the implementer and one other team member, approve a change. The changed artifacts are placed in a current working version directory with appropriate copies of previous artifacts saved.) >>

## Procedures

<< Provide procedures for controlling changes to a software system. Describe in detail the team guidelines for managing your configuration items. Explain what tools (such as SourceSafe or SCC) or policies will be used to document, approve, and make changes to the configuration. If no tools are used, then explain how it will be done manually. How will you document the version number(s) of these tools?

The procedures defined in this section must be consistent with the considerations and procedures defined in other sections of this document. Define who will be in charge of administering the database and making sure that the team follows the process detailed in this part of the plan. Determine in detail what steps each team member must follow in order to checkout and modify an item, what steps are needed to create a new baseline for the project and what approvals are needed from the configuration control board at each point in the process.

This section of the document must contain enough information to serve as a training mechanism of the above described procedures for new team members.

>>

# Software Configuration Auditing

<<Provide a mechanism for determining the degree to which a configuration of the software system mirrors the intended software system. This section describes the process your organization will use to ensure that a delivered version contains the updates intended. Section 3 described the process for identifying and propagating changes to a base line. This section describes the process for determining the compliance with the process in Section 3. >>

$$$