
CONFIGURATION MANAGEMENT PLAN

GuessWho!™

Face-Name Matching Game for Dementia Patients

Submitted to –

Dr Althea Liang

Nanyang Technological University

School of Computer Science and Engineering

Delivered by

Bookies

Joshi Chaitanya Krishna (U1522971F)

Genevieve Lam Wen Qi (U1521863H)

Sharma Vidur (U1522940D)

Yong Chen Feng (U1620913B)

Heng Zhi Guang (U1620660F)

Chen Guanyu (U1621093D)

Identification	3
Purpose	3
1.2 Objectives	3
1.3 Document Overview	3
1.4 Abbreviations and Glossary	5
1.4.1 Abbreviations	5
1.4.2 Glossary	5
1.5 References	5
1.5.1 Project References	5
1.5.2 Standard and Regulatory References	6
Organization	6
2.1 Activities and Responsibilities	6
2.2 Decision Process and Responsibilities	7
3. Configuration Identification	8
3.1 Identification Rules	8
3.1.1 Identification of Configuration Items	8
3.1.1.1 Version Number of Configuration Items	8
3.1.2 Identification of Documents	9
3.1.2.2 Version Number of Documents	9
3.2 Configuration Baseline Management	10
4. Configuration Control	11
4.1 Overview	11
4.2 Change Management	12
5. Configuration Support Activities	13
5.1 Configuration Status Accounting	13
5.1.1 Evolutions Traceability	13
5.1.2 Setting up Configuration Status	13
5.1.3 Configuration Status Diffusion	13
5.1.4 Configuration Status Records Storage	14
5.2 Configuration Audits	14
5.3 Reviews	14

5.4 Configuration Management and Maintenance Plan	15
Appendix	16
Software Change Request (SCR) Form	16

1. Identification

1.1.Purpose

This document contains the Software Configuration Management Plan (SCMP) of the software application GuessWho!™, developed by the team Bookies.

As a facet of the quality control process of the project, configuration management aims to establish, maintain and ensure the integrity of the project throughout the Software Development Life Cycle.

Identifying and controlling configuration items and changes for the project is also part of what configuration management encompasses. This is done by recording the statuses of these configuration items and reporting their change activity.

1.2 Objectives

The objectives of the Software Configuration Management are:

- **Delivery:** Ensures that the correct version of the software is delivered.
- **Traceability:** Ensures that the software being produced, GuessWho!™, meets the stakeholders' requirements.
- **System Integrity:** Ensures that the software and its supporting materials are consistent throughout the Software Development Lifecycle.

1.3 Document Overview

The table below lists the sections that will be documented in this document as well as the contents within each section:

#	Section Name	Section Description
1	Identification	Discusses the abbreviations and technical terms used in the project. Descriptions of the terms are provided in detail.
2	Organization	Describes how the team will be organised to perform configuration management. The duties and jobs of each team member are listed as well.
3	Configuration Identification	Describes how the Software Configuration Items will be identified.
4	Configuration Control	The configuration change management process in the Software Configuration Items of GuessWho!™ will be described here.
5	Configuration Support Activities	The activities that evaluate the degree of compliance towards the procedures and standards in implementing changes to the Software Configuration Items (established within the Software Configuration Management Plan) demonstrated by the development team will be described here.

1.4 Abbreviations and Glossary

1.4.1 Abbreviations

Abbreviation	Description
SCMP	Software Configuration Management Plan
SCM	Software Configuration Management
SCI	Software Configuration Item
SCR	Software Change Request
SRS	System Requirements Specification

1.4.2 Glossary

Glossary Term	Description
Software Configuration Management (SCM)	The discipline of identifying the configuration of a system for controlling changes to this configuration as well as maintaining the integrity and traceability throughout the Software Development Life Cycle.
Software Configuration Management Plan (SCMP)	A reference document for the SCM process that includes the work breakdown structure and description of each section of the process.

1.5 References

1.5.1 Project References

#	Ref. Code	Document Title
1	[RF1]	Dementia in the Asia Pacific Region – Alz. (n.d.). Retrieved 7 September, 2018, from https://www.bing.com/cr?IG=27DC3F982B0547AB8FC683CB71901C5E&CID=0F8FA9448F326A3E28F3A35E8E036BD0&rd=1&h=Y6CYCYwLP8bbWNC_Pk8SPckj2Nf7-Bc7QfKKwxa5h0&v=1&r=https%3a%2f%2fwww.alz.co.uk%2fadi%2fpdf%2fDementia-
2	[RF2]	Wilson, R. S., Kaszniak, A. W., Bacon, L. D., Fox, J. H., & Kelly, M. P. (1982). Facial Recognition Memory in Dementia. <i>Cortex</i> , 18(3), 329-336. doi:10.1016/s0010-9452(82)80031-2
3	[RF3]	What is Dementia http://www.alz.org/what-is-dementia.asp
4	[RF4]	What is Dementia https://www.alzheimers.org.uk/info/20007/types_of_dementia/1/what_is_dementi
5	[RF5]	Clark, L. R., Stricker, N. H., Libon, D. J., Delano-Wood, L., Salmon, D. P., Delis, D. C., & Bondi, M. W. (2012). Yes/No Versus Forced-Choice Recognition Memory in Mild Cognitive Impairment and Alzheimer's Disease: Patterns of Impairment and Associations with Dementia Severity. <i>The Clinical Neuropsychologist</i> , 26(7), 1201-1216. doi:10.1080/13854046.2012.728626
6	[RF6]	Flicker, C., Ferris, S. H., Crook, T., & Bartus, R. T. (1990). Impaired Facial Recognition Memory in Aging and Dementia. <i>Alzheimer Disease & Associated Disorders</i> , 4(1), 43-54. doi:10.1097/00002093-199040100-00005
7	[RF7]	Population Trends. (2016). Retrieved from http://www.singstat.gov.sg/statistics/visualising-data/storyboards/populationtrends
8	[RF8]	Dementia in the Asia Pacific Region: Statistical Appendix. (2006).

		Retrieved from http://www.alz.org.sg/sites/alz.org.sg/files/About%20Dementia%3B%20Statistics%20-%20AsiaPacificEpidemicSept06%20-%20Statistical%20Appendix.pdf
9	[RF9]	Dementia in the Asia Pacific Region. (2014). [pdf] London: Alzheimer's Disease International, London, p.53. Available at: https://www.alz.co.uk/adi/pdf/Dementia-Asia-Pacific-2014.pdf [Accessed 7 Sep. 2018]
10	[RF10]	Elderly Population in Singapore Understanding social, physical and financial needs. (2015). [pdf] Singapore: Lien Centre for Social Innovation, p.1. Available at: https://lcsi.smu.edu.sg/sites/lcsi.smu.edu.sg/files/publications/Elderly-population-in-Singapore.pdf [Accessed 7 Sep. 2018].
11	[RF11]	Satghare, P., Chong, S., Vaingankar, J., Picco, L., Abdin, E., Chua, B. and Subramaniam, M. (2016). Prevalence and Correlates of Pain in People Aged 60 Years and above in Singapore: Results from the WiSE Study. Pain Research and Management, 2016, p.1.
12	[RF12]	Annual Survey on Infocomm Usage in Households and by Individuals for 2017. (2018). [ebook] Singapore: Infocomm Media Development Authority, p.7. Available at: https://www.imda.gov.sg/-/media/imda/files/industry-development/fact-and-figures/infocomm-survey-reports/hh2017-survey.pdf?la=en [Accessed 7 Sep. 2018].
13	[RF13]	Nielsen, J. (1993). Usability engineering. 1st ed. Amsterdam: Morgan Kaufmann, p.135.

1.5.2 Standard and Regulatory References

#	Reference Code	Document Title
1	[STD1]	IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998

2. Organization

Team Bookies will be managing the software application GuessWho!™. The tasks of the Configuration Management will be shared between the Project Manager, Release Engineer, as well as the Quality Assurance Manager & Engineer.

2.1 Activities and Responsibilities

The table below shows the different activities as well as the members responsible:

Task	Responsible Member(s)
SCMP Setup	
Identification of SCIs	Quality Assurance Manager & Engineer
Installation of bug repository tool and setting up of database	Quality Assurance Manager & Engineer
Installation of software configuration repository tool	Project Manager
Set up of documents archive repository	Release Engineer
Project Development Lifecycle	
Export components for modification, test and delivery	Quality Assurance Manager & Engineer
Version and archive version creation	Release Engineer
Verify versions to be delivered and authorise deliveries	Project Manager
Configuration of audit	Quality Assurance Manager & Engineer

Monitor configuration records	Quality Assurance Manager & Engineer
Management	
Manage versions and archives	Release Engineer
Archives for documents	Release Engineer
Archives for software configuration	Release Engineer
Quality report management	Quality Assurance Manager & Engineer

2.2 Decision Process and Responsibilities

The list of decision process and responsibilities made through the Configuration Management Plan are documented in the table below:

Activity	Responsible Member(s)
End of an Activity of the Project	
Performing configuration freeze	Release Engineer
Presenting configuration state of components impacted by activity	Release Engineer
Presenting documentation state of components impacted by activity	Release Engineer
During Configuration Management Process Audit	
Perform configuration management process audit	Project Manager
Presenting records of configuration management process	Release Engineer
Presenting quality records of the configuration management process	Quality Assurance Manager & Engineer
Presenting the records of the documentation management process	Release Engineer

3. Configuration Identification

3.1 Identification Rules

The identification of items that have to be controlled throughout the Software Development Life Cycle of this project and the creation of the identification scheme will be discussed in this section.

3.1.1 Identification of Configuration Items

The purpose and development phase of Software Configuration Items (SCIs) will define the format for identifying SCIs. The SCIs that will be composed to form our product will consist of:

- Source Code (HTML, JavaScript Files, External Libraries)
- Database Information

The naming of the Source Code files will be in the format:

[Function Name]_[Version Number].[File Format]

Moreover, source code files will also be organized into different folders, mainly front-end functions and pages, and back-end services and controllers respectively.

3.1.1.1 Version Number of Configuration Items

Assigning unique version numbers to the software reflect the changes made as well as the severity of the changes made.

The version number assigned will increase by *0.1* for each minor change, and by *1.0* for each major change. The definitions of the changes are listed below:

Major	Changes made to the main features of the Software Application.
Minor	Bug fixes, changes, or additions of minor features to the Software Application that do not affect the main features. Increments of the Minor version number will be reset to zero once the Major version number increments by <i>1.0</i> .
Revision	Change or addition of features that are of least significance to the Software Application. Examples include formatting or optimization of code. Increments of the Revision version number will be reset to zero once the Minor version number increments by <i>0.1</i> .

3.1.2 Identification of Documents

The purpose and development phase period of documents will define the format for identifying documentation. The documents that form our product will consist of the following:

Project Proposal
System Requirements Specification
Quality Plan
Project Plan
Risk Management Plan
Design Report on Software Maintainability
Configuration Management Plan
Change Management Plan
Release Plan
Test Plan

The naming of Documents will be in the format:

[Document Name]_[Version Number].[File Format]

Moreover, the documentations will also be organized into different folders, based on their development phase of the Software Development Life Cycle.

3.1.2.2 Version Number of Documents

Just as in 3.1.1.1 *Version Number of Configuration Items*, documents will be assigned version numbers as well to reflect the changes made and the severity of the changes made.

It should be noted that Revision Version Numbers will not be included in the Version Number for Documents.

The definitions of the changes are listed in the table below:

Major	Significant amendments made to the documentation. Examples include content modification. Amendments made must be approved by the Project Manager.
Minor	Amendments made to the documentation which is not as significant. These include formatting of documentation, or correcting spelling and grammatical mistakes. Amendments of such a nature do not have to be approved by the Project Manager. Increases in the Revision version number will be reset to zero once the Major version number increases by <i>1.0</i> .

3.2 Configuration Baseline Management

Team Bookies has established a number of baselines in order to ensure that the Project for GuessWho!™ is carefully scrutinised and kept under control.

Serving as reference points in the Software Development Lifecycle, these baselines play an important role as they mark the completion and formal approval of a set of Configuration Items. Changes to the SCI after a baseline has been established have to be made formally and must follow the controlled procedures documented.

These baselines identified by the team includes the following:

Baseline	Period of Establishment	Components
Functional	After requirements have been reviewed, when first major version has been released.	Project Proposal System Requirements Specification Project Plan
Design	After Design Review phase	Class Diagram Use Case Model
Product	After User Acceptance Testing	Source Code Test Plan Test Cases Quality Plan
Operational	After deployment of Software Application	Source Code Test Plan Test Cases

4. Configuration Control

4.1 Overview

Change management within the Software Configuration Items during the Software Development Lifecycle is a part of configuration management. By having proper control, changes made to the system are done with acknowledgement and the consent of the team (the Project Manager in particular), and that unsupervised changes are avoided.

Configuration control is made up of the steps listed below:

- Recognising that a change is necessary
- Discuss the change request submitted
- Approval or Disapproval of the change request
- Verification, Implementation and Release of the change request

Any developer is allowed to propose changes. However, changes must be documented and submitted through the Software Change Request Form. The information below must be included when documenting the change proposal:

- Explanations regarding the suggested change should be implemented
- Parts of the system that will be affected by the change
- Sections of code that have to be updated and other resources that may be required.

In order to submit change requests, the Software Change Request form must be filled up. This form must be reviewed by the Lead Developer and the Quality Assurance Manager & Engineer before the change can be approved. The team members working on the affected portions of the Software Application will then carry out the change in a different branch once the change is approved, minimizing the effect of the change on the current branch.

In order to check that the new code has no bugs, test cases will be performed on the new code. After the code is tested, the branch will merge with the original branch. In this way, the original branch will have the latest, working code.

4.2 Change Management

If changes have to be made to satisfy the requirements of the project, developers can opt to submit change requests (SCRs). Minor SCRs that require only a revision of different portions of the Software Application should be implemented using a different branch and must be tested before merging it with the original branch.

In order to supply the details about the implemented change, proper commit messages should be used. On the other hand, the branch name should identify which portion of the System Application is being changed. In addition, the branch name should also identify the developer implementing that change.

Developers must fill up a detailed SCR form and submit it to the Project Manager, Quality Assurance Manager & Engineer as well as the Release Engineer when proposing major SCRs that involve changes to the architecture of the System Application.

In order to ascertain the validity of the proposed change and the feasibility of implementing such a change, the form will be evaluated by the Quality Assurance Manager & Engineer and the Release Engineer.

Developers will then create a new branch based on the SCR form to begin implementing the changes on after the change has been assessed and approved. After tests on the change indicate there are no bugs, the changes made will then merge and integrate with the original branch, and the software version will be modified based on the impact of the change (Major, Minor, or Revision). After which, the updates must be passed throughout the development team and the respective repositories, and commit messages from the new branch must clearly state the changes made as well as the reason for making those changes.

A template of the SCR Form used in this project can be found in the Appendix of this documentation.

5. Configuration Support Activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process of documenting, storing, maintaining, and reporting the statuses of Software Configuration Items during the Software Development Life Cycle. In order to ensure traceability, all SCIs and relevant documentation must be tracked. To track changes in the system, version numbers of the SCIs are used. They are responsible for:

- Elements that have to be tracked and reported for baseline changes.
- The types of Status Account Reports that are to be generated and the frequency of generating such reports.
- The collection, storage and reporting of information.
- Access control to the Configuration Management data of this project.

5.1.1 Evolutions Traceability

Software Configuration Item (SCI)	Traceability
Documentation	Version number within the documentation and the documentation file name will be stored. Changes made to the documentation will update the version number. The documentation will be saved with a different file name, based on version number.
Source Code	Commit ID and Message/Description
Software Change Request	Request ID
Design Diagrams	Version Number within file name
Test Cases	Version Number within file name
Version Description Document	Version Number within file name

5.1.2 Setting up Configuration Status

In light of manpower constraints within the development team, the Lead Developer will take over the role of the Software Configuration Manager and write the Version Description Document.

Additionally, the Lead Developer will set up the state of all versions of each Software Configuration Item with the label, version and date of creation of the Version Description Document.

5.1.3 Configuration Status Diffusion

Both the Quality Assurance Manager and the Lead Developer will write the Version Description Document. The Project Manager must vet and approve it in order for it to be considered valid.

5.1.4 Configuration Status Records Storage

The records of each SCI are stored in a configuration folder, which contains the following:

- Software Change Requests, sorted by Request ID
- Software Documentation
- Version Description Documents
- Configuration States sorted chronologically

5.2 Configuration Audits

The majority of the Configuration Audits in this project are baseline audits. They involve ascertaining the baseline that decides the agreed attributes and description of the software application, GuessWho!™.

The most recent status of the SCIs is examined so as to ensure that each of their performances are consistent with specifications mentioned within the System Requirement Specifications. The Quality Assurance Manager and Engineer will conduct formal examinations of test results of the functional configuration of SCIs in order to meet the identified requirements.

In order to promote learning from past functional configuration audits, a standardized checklist can be used during the functional configuration audit. By doing so, each SCI can be made to be consistent with specifications determined and documented within the SRS.

5.3 Reviews

The baseline for the project is explained by the audits mentioned. Because the future states of the project can only be determined by knowing the current state, the audits form the foundation for changes in the future to be made.

A project management tool that helps to set up these baselines is known as a review. The table below lists the roles and responsibilities of reviews:

Roles	Responsible Member(s)
Select the objectives and Software Configuration Items for review	Lead Developer
Planning and deciding a schedule for the review based on procedures	Quality Assurance Manager & Engineer
Come up with procedures to document the deficiencies within the project as well as the solutions to these deficiencies	Quality Assurance Manager & Engineer
Comes up with approval criteria	Lead Developer

5.4 Configuration Management and Maintenance Plan

In this project, the Configuration Management plan acts as a reference and documentation for the process of the Software Configuration Management. Throughout the whole Software Development Lifecycle, it will be updated. This way, it can be used to help define how changes applied to identified requirements will be performed.

The monitoring of these plans is to be done by the Quality Assurance Manager & Engineer. To make sure that everything is going according to schedule, and updates have to be performed according to the Software Configuration Management plan every week. The Quality Assurance Manager & Engineer must assess and approve all changes to these plans.

The latest plan is saved with a newer version number, once approval for the plans have been granted. The configurations folder will store all the old plans, which can be accessed to help facilitate the project schedule.

Appendix

Software Change Request (SCR) Form

Name of Initiator:	Position:	Reference Number:
Contact Details:	Date:	
Description of Change:		
Priority (Critical/High/Medium /Low):		
Justification for Change:		
Software Configuration Item Affected:		
Documents Changed:		
Preliminary Assessment:		
Impact Assessment:		
Remarks:		
Decision (Approved/Reject/Hold):		
Signature of Project Manager:	Date:	