SeniorConnect Project
Process Deifinition
CMMI Level 2 Requirements

Version 2.0

06/11/2015

GruFamily Project Team
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798

Revision History

Name	Date	A*MD	Reason For Changes	Version
Li Yishan	30/10/2015	A, M	Draft the plan	v1
Mao Huiqi	06/11/2015	A, M, D	Final revision and modification of the plan	v2

Table of Contents

1. Introduction

This document serves to record the processes and the management of the SeniorConnect system thus demonstrates compliance with CMMI level 2. These processes and management will then be reviewed to improve their effectiveness.

The SeniorConnect system was developed for SCE with the intention of creating a user-friendly mobile application for Senior Citizens to connect with each other socially and encourage their participation in social activities.

2. Software Configuration Management

2.1. Process Goal

Software Configuration Management (SCM) aims to ensure the integrity of work products (in the following sections, referred to as configuration items) throughout the software development life cycle. This is due to the possible significant results from changes of certain work product because other work products may be directly or indirectly affected. Therefore, SCM is conducted to ensure the consistency of work products (including documents, code, hardware and other forms of items) for each release.

2.2. Commitment to Perform

A project-specific configuration management team is formed. According to the SC Configuration Management Plan, Software Configuration Management (SCM) team is formed by project manager and QA manager. Lead developer, front-end engineer and back-end engineer, regarded as development team, together with release manager should assist SCM team while SCM team is involved in SCM processes. The detailed description of SCM organization and team’s responsibility is presented in Configuration Management Plan.

The SCM plan is finalized by SCM team and controlled under configuration management. SCM plan is corporately fulfilled by SCM team, release manager and development team.

During the whole project life cycle, QA manager is in charge of SCM of QA documents; development team is in charge of SCM of source code, libraries, code documentation, and design diagrams; project manager is responsible for SCM of project planning relevant documents and other configuration items.

Major changes to the baselines of configuration items are approved by project manager. If the changes relate to major releases, the approval from release manager is also needed.

2.3. Ability to Perform

SCM Team is formed to ensure the integrity of work products; team members are trained to adhere to SCM plan thereafter. Relevant personnel, developer team and release manager, should answer requests from SCM team to assist SCM tasks.

Baselines of each configuration items are ready at pre-defined stage of project life cycle. Future improvement is built upon the baseline to form the current configuration. The changes from the previous baseline to a current baseline are recorded in traceable details.

Configuration management system is built to keep track of changes and maintain alignment of all configuration items. These systems include: documents for team wiki pages, code for SVN.

QA team is ready to review configuration items at each milestone of the project, and SCM team is ready to take corrective actions to issues raised during the review.

2.4. Activities Performed

Build the configuration management software system and ensure a high availability, e.g. 24/7 accessibility.

Identify the items that should be kept as configuration items. Criteria of selecting additional items are listed in Appendix I. Criteria of Selection of Configuration Items.

Create the baselines of configuration items when the first approved versions of configuration items are ready.

Label configuration items properly based on the nature and versions of the items according to Configuration Management Plan.

Make periodical backup of current configuration items and label it properly, e.g. timestamp.

Control changes by holding a separate working set of configuration items from baselines of configuration items. Changes before baseline are modified and tested upon working set. Baseline system is maintained unchanged until a change request is approved.

Track changes use existing configuration management, as described in the previous section.

Changes are made for every necessary improvement. The reason can be a new requirement is fulfilled, a requirement is changed, or a defect is found in the previously established work product. After a new change is fully tested and to be applied to baseline items, a change request should be raised, analyzed, and approved/rejected. Further validation is conducted to ensure complete implementation of changes of change assignment.

Audit configuration items to ensure verification of configuration items are complete, consistent, and accurate.

2.5. Measurement and Analysis

Organize configuration items physically in SCM systems according to the nature of configuration items. Configuration items should be categorized clear and intuitive.

Name each configuration item with unique identifier according to the naming scheme described in Configuration Management Plan.

Control the SCM system accessibility by ranking each member with respective rights and limitations. SCM system should be maintained with high availability with a fault time of less than 0.1% of the time.

Backing up baselines of configuration items immediately after every change request is implemented fully, and after every major release. Every backup should be kept for 6 months at minimum.

Override released set of configuration items by working set is never allowed without project manager and release manager's approval. Time consumed for roll-back action should be recorded and summarized.

Make current configuration of configuration items known to all relevant personnel, including project manager, QA team, and development team immediately after each each milestone of the project becomes a new baseline or is released.

2.6. Verifying Implementation

SCM plan should be ready and approved by project manager when the initial baseline of each configuration item is ready.

Each document maintained and tracked via team wiki should be clearly labeled.

SVN should maintain at least two branches, one is working branch for working items and another is released branch for baseline items. All developer members should know the organization information of SVN. Commits history should be kept automatically to ensure no false commit to released branch is made.

Separate physical storage should be used to store backup configuration items, for this project, the items are versions of SVN and documents on team Wiki.

History of changes to every configuration item should be maintained and traceable in SCM system via SVN and team Wiki. Each update should be recorded in change history including content, reason, and sections changed.

Configuration management audit should be conducted at each major release. Multiple audits could take place for one release. *SCM Process Audit Form* in Change Management Plan should be used to record the issues, findings and other relevant information during the auditing. Issues and findings are to be resolved and status of issues should be updated. All issues should be settled before release.

3. Software Quality Assurance (SQA)

3.1. Process Goal

The purpose of software quality assurance is to evaluate the individual process during software development process with reference to compliance regulations according to Quality Plan. For every non compliance-compatible issue, corrective actions shall be provided and conducted. Quality assurance provides useful suggestions to the quality of work products and working processes to each of the project team and management. The main goal of software quality assurance is to ensure all processes and work products observe standard quality criteria and quality control process.

3.2. Commitment to Perform

QA team is formed to serve as a supervisor working independent from the project team to evaluate work products and processes.

Project manager and development team should assist QA tasks by providing necessary documents and work products for assessment.

When a non compliance-compatible issue is discovered, corrective actions should be taken by personnel involved under supervision by QA team.

3.3. Ability to Perform

SQA plan is developed by QA team, revised by the whole project team, and approved by project manager.

SQA checklists and audit forms are provided by SQA plan.

Software Requirement Specification documents including most up-to-date functional, non-functional, and interface requirement are ready and used as reference to assist QA tasks.

3.4. Activities Performed

Evaluate software development processes against standard processes defined in SQA plan, and provide QA feedback in terms of the processes evaluated.

Evaluate work products against requirement documents like SRS using checklists in SQA plan, and provide QA feedback for selected work product.

The feedback should raise any non compliance-compatible issues found and provide quality evaluation. Non compliance-compatible issues should be documented and communicated to relevant personnel. Each issue raised should have a responsible person in charge of organizing corrective actions.

QA teams should check issues status periodically to ensure all issues raised are under progressing or addressed. Changes resulting from noncompliance issues should be properly made and maintained in the document. Re-reviews should be conducted by QA team to evaluate the changes made with respect to the issues. Only after issues are addressed, QA teams can close the issue case but maintain the issue records.

3.5. Measurement and Analysis

Records of noncompliance issues should be maintain to provide evidence for analysis of overall adherence to the SQA plan at a later stage.

Records of non compliance-compatible issues for each work product should be maintained separately in the form of review form and audit forms provided in QA plans for future reference and tracking.

Records of non compliance-compatible issues currently opened and resolved should be maintained. It is a good practice to maintain the number of non compliance-compatible issues (in terms of work products) opened less than 2 overall.

Cost and time required to resolve the non compliance-compatible issues are recorded in the review/audit form to reflect the loss from non compliance-compatible actions. It could be indirectly recorded as re-review time from date of review.

Records of reviews conducted for each process or work product should be maintained. It is a good practice to maintain this number less than 3 unless severe noncompliance issues are discovered.

Records of issues and details of issues found in each review meeting should be documented and maintained.

3.6. Verifying Implementation

Process Audit Form (III-E in SQA Plan) should present for each QA review of process; review forms and checklists should present for each QA review of work product.

Non compliance-compatible issues are properly documented and monitored periodically. Changes resulting from non compliance-compatible issues should be recorded in the change history of each work product via SVN or stated in the new work product if applicable.

All other documents required in SQA plan should present and satisfy the required level of quality if a process or a work product passes QA review and audit.

4. Software Project Tracking & Oversight 4.1. Process Goal

The purpose of software project tracking and oversight is to ensure project manager and the team has clear understanding of the project progress. In the case where project plan and schedule deviated significantly from the plan, the team can take corrective action.

4.2. Commitment to Perform

Project plan is documented by project manager and approved by the management before the project starts.

After the project team is formed, project plan is modified to take proper estimation, buffer time, and justified feasibility into consideration. Then the project plan is formally endorsed by the team, and approved by project manager.

Project plan is to be followed by project manager, development team and QA team.

Project manager should control the pace of the project and take actions to ensure the team meets project timeline and milestones according to project plan.

The team should conduct corrective actions following project manager’s arrangements if the progress deviated from the plan to a significant extent.

QA team should review project plan and project progress.

4.3. Ability to Perform

Project plan and project timeline are initially prepared by project manager, modified to great detail by project team, and approved by project manager. Team members agree to the project plan.

Buffer time is provided to provide time to handle unexpected issues and ensure the adherence to planned project timeline.

Audit and review processes are scheduled to detect the deviation from project plan, ensure corrective actions are taken properly and maintain all relevant documents to be traceable.

4.4. Activities Performed

Monitor actual project progress against planned project timeline. The project progress includes the status of project components and quality of project components.

Monitor resources used for the project and compared against the plan. Make arrangement to ensure correct amount of resources are used until current stage.

Monitor the deliverables and non-deliverable work-in-progress items for each milestone and project phase. Ensure the items are meeting stakeholders' requirements and communication such as meetings are conducted at scheduled time.

Monitor the commitment of project team members to ensure the knowledge and skills of project team members are utilized to a maximum extent.

Conduct progress review with the team every week and the findings during the reviews should be documented and traceable. Conduct milestone review with the team whenever a milestone is reached. In the case where deviation from project plan is discovered in the review, the team should reach a consensus on corrective action; after project manager's approval this action is properly documented and executed. Project manager should monitor the progress and effects of the corrective action. The result of corrective action is documented.

4.5. Measurement and Analysis

It is a good practice to maintain the total number of deviations from plan less than 10 as a team.

It is a good practice to control the delays of deliverables within 3 days after the due date. Total delay resulting from deviation from project plan should be recorded.

Any changes to the project plan due to delaying of progress or over-budgeting should be recorded in detail including the reasons deriving the new timeline. It is highly recommended that the total number of changes due to execution deviation should be below 3.

For any deviation, the corrective period maximizes 2 weeks, and amount of delay for each deviation is to be documented with great detail including the time allocation for the corrective period.

The status of each project deliverable should be made available to the project team during the weekly review meeting and at each milestone review meeting. QA team should also be informed in order to prepare reviewing. Release manager is to be informed in the case of a release.

4.6. Verifying Implementation

Management progress review should be properly documented in the Management Review Form.

Any deviation is properly recorded in Management Review Form, together with corrective action to the deviation and the progress or the result of the corrective action.

Changes to project plan due to deviation should be properly documented in the project plan. Project manager should approve the changes, after which project plan should be versioned as a new baseline.

Meeting minutes should be kept for involvement with stakeholders.

Audit form should be filled in to ensure progress reviews are conducted.

5. Software Project Planning

5.1 Process Goal

The purpose of this process area is to make a project plan that can be used as a guideline for future project activities. It documents all managerial aspects of a project that are required to execute it successfully within its constraints. The project plan will also be used as a baseline for project progress.

5.2 Commitment to Perform

5.2.1 Project manager agrees to make a project plan with documentations.

5.2.2 Project plan needs to be reviewed by project manager, QA manager and lead developer to assess its feasibility.

5.2.3 If there is any change needed in the project, relevant groups will take part in the change analysis and review session, and give related input. Project plan cannot be changed without a formal meeting with stakeholders, including team members and customer.

5.3 Ability to Perform

5.3.1 Resources in terms of knowledge base and time are allocated to project planning. Project manager cooperates with experienced team members to develop the project plan.

Relevant templates such as Gantt chart and critical path are available to use. SmartSheet is an online tool to aid the process planning procedure.

5.4 Activities Performed

5.4.1 The scope of the project was estimated as well as work products and task attributes. The scope was estimated based on the expected number of requirements, complexity of requirements and available resources.

5.4.2 Selected software development lifecycle (Agile) for the project based on the nature and property of the project.

5.4.3 Estimated effort, duration and team size in this project.

5.4.4 Developed project timeline, including the following:

- Project major milestones
- Assumptions for making schedule
- Task timeline (start date, duration)
- Task dependencies
- Person in charge
- Other constraints on tasks

5.4.5 Analyzed project risks and made proper documentation

5.4.6 Provided a list of deliverables and work-in-progress work products at each milestone.

5.4.7 Allocated resources at each phase of project life cycle. Set time for self-learning when necessary.

5.4.8 Planned involvement of relevant stakeholders at certain time in project execution

5.4.9 Obtained commitment from team members to the project and got the project plan approved

5.4.10 When requirement was changed or infeasible plan was discovered, changes were made to the project plan.

5.5 Measurement and Analysis

5.5.1. Amount of time required for the project as a whole should be similar to projects of similar scope.

5.5.2. Number of changes happened to the project plan and reasons for the changes should be recorded.

5.5.3. Involvement of relevant stakeholder should not be fewer than two times during the project.

5.5.4. Project deliverables for each phase should not violate the dependencies among deliverables.

5.5.5. Amount of time elapsed from the first draft of project plan to the approval of project plan should be recorded. The time for approval is usually kept within 1 day.

5.6 Verifying Implementation

5.6.1. High level Work Breakdown Structure should be presented and become the basis of project planning.

5.6.2. Justification of estimates regarding budget, timeline, and resources of project should be documented in a separate document.

5.6.3. Project plan should be ready for review to assess its feasibility and completeness. The reviews should be documented in Review Form.

5.6.4. Commitment obtained from the team should be properly recorded by holding a meeting where the group reaches a consensus.

5.6.5. Historical data for similar projects should be kept separately as supportive documents for the estimation of current project.

6. Requirement Management

6.1 Process Goal

The purpose of this process area is to establish common understanding between customers and the project team of customer requirements in the project. It ensures the product meets customers' needs. Agreement with customer is the basis for planning and managing the software project. It also ensures alignment between requirements, project plan and work products.

6.2 Commitment to Perform

The project follows requirement management policies allocated to the software in the organization. The project needs to identify inconsistencies between the requirements and the project plans and work products. The policy specifies:

- The allocated requirements are documented in the form of use case diagrams, use case descriptions and SRS.
- The requirements are reviewed by lead developer and approved by project manager.
- Any changes occurred to requirements are kept track of and use case review is led by QA team.
- The software plans, work products, and activities are changed to be consistent with requirement changes

6.3 Ability to Perform

6.3.1. Resources in terms of time and manpower are provided for managing the allocated requirements. Individuals who have experience in requirement elicitation and management domain are allocated to this job.

6.3.2. Requirement management has been previously planned and scheduled in the software development lifecycle

6.3.3. Relevant templates such as Use Case Description template is prepared beforehand.
6.3.4. Members in the team are trained to be professional in requirement management in order to perform their own requirement management activities.

6.4. Activities Performed

6.4.1. Requirements were elicited from customers by interviewing them for their needs and providing reasonable features to test their preference. Measures used include market research, face-to-face interview, investigation of similar products, customer survey, etc.
6.4.2. Functional requirements were properly documented in use case description, and all changes were maintained in the form of use case subsequently.
6.4.3. Non-functional requirements were properly documented in software requirement specifications.
6.4.4. All requirements were reviewed by developer team to investigate the feasibility, checked by personnel in charge of requirement management, and confirmed with customers before proceeding to design.
6.4.5. Incorrect, incomplete, and missing requirements were identified and became the basis of requirement review. They should be corrected in the next revision of requirement and the changes happened should be properly documented by version control. In review of each requirement, Software Requirement Review Checklist (5.13.3 in SQA Plan) should be used as criteria for checking.
6.4.6. In case of changes to requirements happen, maintain relationships among requirements, project plans, and work products, and ensure alignment among them.

6.5. Measurement and Analysis

6.5.1. After activities mentioned above are performed, work products include use case diagram, use case description, and software requirement specifications.
6.5.2. Use case description should contain functional requirements and some nonfunctional requirements. Software requirement specification should contain functional requirements, non-functional requirements, and interface requirements.
6.5.3. Each requirement should be properly labelled and uniquely identified.
6.5.4. Status of each of the allocated requirement should be tracked and made known to project manager and QA team at each milestone of project. Status of each requirement should be compared against the planned progress and accepted when it is not delayed.
6.5.5. Total number of changes, approvals, and amount of time required for changing existing requirements should be tracked. Total number of changes should not be more than 5.

6.5.6. Number of changes initiated by customer should be tracked to analyze the effectiveness of requirement elicitation. Total number of changes initiated by customer should not exceed 3.
6.5.7. Number of changes initiated by development team or from within the project team should be tracked to analyze the capability of the team in implementing the required features and capability of self-correction of the team. This number should not be more than 3.
6.5.8. Number of chain changes incurred to other requirements, project plan, and work product resulting from change of certain requirement is tracked to analyze the coupling and granularity of each requirement.

6.6. Verifying Implementation

6.6.1. Before a requirement is accepted, it should be evaluated against traceability, completeness, consistency with others, uniqueness, testability, and relation to business value. Project team only records accepted requirements and they are kept as set of use cases and SRS.
6.6.2. Requirements are reviewed to determine whether they are feasible and appropriate for implementation in software. Inappropriate requirements are removed from use cases and SRS.
6.6.3. Requirements are reviewed as a whole and against designs, implementation, and test plans to determine bidirectional traceability and alignment, possibly in Functional Configuration Audit and Physical Configuration Audit.
6.6.4. Each changes, regardless of initiator and nature of change, should be properly documented by keeping track of changes happened to each use case. The number of changes made, initiator, proper current version, and content of change should be recorded in the note of each use case.
6.6.5. All review results from 6.6.1 to 6.6.3 should be kept in Software Requirement Analysis Process Audit Checklist (5.14.1 in SQA Plan).
6.6.6. Final deliverable should include a set of approved use cases, software requirement specification, and a set of changes made over time. Project manager, QA manager, and lead developer should approve this final deliverable.

Appendix I. Criteria of Selection of Additional Configuration Items

Additional Configuration Items			
General Information			
Configuration Item			
Date			
ID	Y/N	Aspect to check	Comments
1		Work product is used by multiple groups	
2		Work product is expected to change over time	
3		Work product is related to other products and change in this product may result in riddle changes in other products	
4		Work product is part of the deliverable and critical to the project	
5		Work product requires collaboration of team members	
6		Work product may have various copies and versions available	

Endorsed by:

Project Manager   Signature:__________________