EX-SELL

CMMI Level 2 Definition

Version 1.0 approved

Prepared by Zenith

Yiu Hong Sum	U1621435G
Lim Kian Hock Bryan	U1620949L
Cheng Yang Zhen	U1521618K
Lim Yan Jun	U1622311B
Soong Jie Ming	U1521535B
Chen Zhenni	U1622603D
Chang Keat Lueng Aaron	U1622360F

29th March 2018

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	Initial Template	Aaron	20/03/2018
0.20	Add Process Areas	Aaron	23/03/2018
1.00	Revised Edition	Hong Sum	29/03/2018

Table of Content

1. Introduction

Capability Maturity Model Integration (CMMI) is a framework of best practices. The purpose of CMMI is to provide guidance for setting up and improving an organization’s processes areas.

CMMI model contains five maturity levels – Initial, Managed, Defined, Quantitatively Managed and Optimizing. Each of the maturity levels consist of a predefined set of process areas. This document will focus on CMMI level 2 and provides description on the key activities of each process areas. CMMI level 2 ensures that all work products are appropriately controlled. Therefore, CMMI level 2 satisfy the expectations of the process descriptions, standards, and procedures.

The following list shows the six key process areas of CMMI level 2:

1. Requirements Management
2. Software Project Planning
3. Software Project Tracking & Oversight
4. Software Subcontract Management
5. Software Quality Assurance
6. Software Configuration Management

2. Process Areas

2.1. Requirements Management

2.1.1. Goal

The goal of Requirements Management process is to define procedures and establish a requirement document based on a common understanding between the customer and the software project requirements.

2.1.2. Key Practices

1. **Obtain Requirements**

   a. **Understand Requirements**

      i. A proposed requirement will need to be established from a relevant provider. If the requirement received is from an inappropriate provider, the requirement will be rejected.

      ii. The requirements are evaluated on the following measures:

         1. Clearness
         2. Completeness
         3. Consistency
         4. Identifiable Uniqueness
         5. Verifiability
         6. Traceability

      iii. Software Requirement Specifications (SRS) will be done based on the compiled requirements.

   b. **Obtain Commitment to Requirements**

      i. When there is a new requirement, or an existing requirement change, the team should assess and evaluate the impact on the participants of the project.

      ii. Before committing to any requirement change, participants of the project should negotiate and note down the changes.

      iii. A revised SRS will be done based on the new requirements.

2. Manage Requirements

a. Manage Requirements Changes

i. Document all requirement changes in the change history document with the reason for each change. This is to ensure that new changes are easily traceable.

b. Maintain Bidirectional Traceability of Requirements

i. Traceability can be established when the requirements are managed appropriately. This is to ensure that source requirements are completely documented, and its lower level requirements can be traced back to the source.

c. Ensure Alignment Between Project Work and Requirements

i. Constantly review the project work and requirements to ensure the consistency. If there are any inconsistencies, the source and rationale will need to be identified. Actions to amend the inconsistencies will need to be initiated when necessary.

2.2. Software Project Planning

2.2.1. Goal

The goal of Software Project Planning process is to establish reasonable plans based on realistic estimates to perform the software engineering on.

2.2.2. Key Practices

1. Establish Estimates

a. Estimate the Scope of the Project

i. Create a work breakdown structure (WBS) for the different components of the project.

ii. Identify the work packages in detail for the estimation of project tasks and schedule.

iii. Identify product components that will be externally acquired.

iv. Identify the work products that will be reused.

b. Establish Estimates of Work Product and Task Attributes

i. The estimates should be consistent with the project requirements to determine the project’s effort, cost, and time needed.

ii. Methods for determining size and complexity should be based on validated models or historical data such as lines of code and function points for software.

c. Define Project Lifecycle Phases

i. The project lifecycle phases will need to be defined based on the scope of requirements and the estimates for project resources.

ii. Estimate Effort and Cost

d. Estimate the project effort and cost by collecting historical data.

i. Supporting infrastructure such as engineering environments and tools are taking into consideration when estimating effort and cost.

2. **Develop a Project Plan**

a. Establish the Budget and Schedule

i. The project’s budget and schedule are based on major milestones, schedule assumptions, constraints and task dependencies.

b. Identify Project Risks

i. Potential risks are identified and analyzed to determine the impact and probability of occurrence.

ii. Document the risks in the Risk Management Plan.

iii. Review the risks with stakeholders and determine the correctness of the risks.

iv. Revise the risks if necessary.

c. Plan Data Management

i. Establish procedures to ensure privacy and security of the data so that only authorised users will have access to the data.

ii. Access to information should be simple so that information can be easily traceable.

d. Plan the Project's Resources

i. Define project resources to perform Project Activities Builds on the initial estimates. This is to ensure efficient operations during the project execution.

e. Obtain Required Knowledge and Skills

i. Identify the required knowledge and skills needed to support the execution of the project.

ii. Assess the knowledge and skills of the staffs available.

iii. If the existing knowledge and skills are not sufficient, external or in-house training will be performed.

f. Plan Stakeholder Involvement

i. Identify stakeholders with the expertise that will be able to contribute to the software development lifecycle activities.

g. Establish the Project Plan

i. Documented a plan that addresses all relevant planning items, this is necessary to achieve the mutual understanding required before executing the plans.

ii. The project plan should include the following

1. Project lifecycle
2. Technical tasks
3. Budgets and schedules
4. Milestones
5. Risk identification
6. Resources needed
7. Skill requirements

3. Obtain Commitments to the Plan

a. Review Plans that Affect the Project

i. Record and review all plans that will affect the project to better understand the project commitments.

b. Reconcile Work and Resource Levels

i. Reconcile the differences between the estimated and available resources.

ii. Plan out ways to increase productivity like outsourcing work or negotiating with stakeholders for more resources.

c. Obtain Commitment Plan

i. Negotiate commitments with relevant stakeholders.

ii. All commitments must be documented to ensure a consistent understanding between stakeholders and to facilitate better documentation.

2.3. Software Project Tracking & Oversight

2.3.1. Goal

The goal of Software Project Tracking & Oversight process is to establish adequate visibility of actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.

2.3.2. Key Practices

1. Monitor the Project in Accordance to the Plan

a. **Monitor Project Planning Parameters**

i. Periodically monitor progress of the project and compare the predicted timing in the project pan and the actual timings for the milestones.

ii. Periodically measure the project's cost and expended effort and compare them with the planned estimates in the Project Plan.

iii. Take corrective action if needed.

b. Monitor Commitments

i. Identify commitments that have not been satisfied and document the results of the commitment reviews.

c. Monitor Project Risks

i. Periodically review the documentation of the risks in the Risk Register.

ii. Take corrective action if needed.

iii. Revise the documentation of the risks when necessary.

d. Monitor Data Management

i. Periodically identify and review data management activities.

ii. Document significant issues and their impacts.

e. Monitor Stakeholder Involvement

i. Periodically identify and review the status of stakeholder involvement.

ii. Document significant issues and their impacts.

f. Conduct Progress Reviews

i. Regularly communicate the status of work progress to relevant stakeholders.

ii. Identify, document, and rectify significant issues.

iii. Refer to the Configuration Management report as a guideline for more information on how to make changes when necessary.

g. Conduct Milestone Reviews

i. Conduct reviews at during milestones in the project's schedule

ii. Review the plan, status, and risks of the project.

iii. Identify and document significant change and decisions.

2. Manage Corrective Action to Closure

a. Analyze Issues

i. Collect information on issues from reviews.

ii. Determine the corrective actions necessary to address the issues.

iii. Rectify the issues.

b. Take Corrective Action

i. Determine and document the appropriate actions by modifying requirements or revising estimates and plans to address the issues.

ii. Inform stakeholders about the actions to be taken and get the confirmation to proceed from them.

c. Manage Corrective Actions

i. Analyze results of corrective actions to determine their effectiveness.

ii. Document the corrective actions.

2.4. Software Subcontract Management

2.4.1. Goal

The goal of Software Subcontract Management process is to manage the acquisition of products from suppliers effectively.

2.4.2. Key Practices

1. Establish Subcontract Agreements

a. Determine Acquisition Type

i. Determine the type of acquisition such as purchasing commercial off-the-shelf products (COTS), through a contractual agreement or from an in-house vendor.

a. Select Suppliers

i. Select subcontractors based on their ability to meet the specified requirements of the project.

ii. When selecting COTS products, evaluate them based on the cost of the COTS products, security requirements and the benefits.

b. Establish Subcontractor Agreements

i. Review the supplier agreement to reflect the project's requirement if necessary.

ii. Ensure that all subcontractors and stakeholders understand the agreement and agree to all requirements.

iii. Document the subcontractor agreement.

2. **Satisfy Subcontractor Agreements**
    a. **Execute the Subcontractor Agreement**
        i. Monitor subcontractor progress and performance.
        ii. Conduct reviews with the subcontractor to improve the performance and relations.
    b. **Accept the Acquired Product**
        i. Verify and accept the acquired products that satisfy their requirements based on project requirements.
        ii. Integrate the finished product into the project.

2.5. Software Quality Assurance

2.5.1. Goal

The goal of Software Quality Assurance process is to provide management with appropriate visibility into the process being used by the software project and of the products being built.

2.5.2. Key Practices

1. **Objectively Evaluate Processes and Work Products**
    a. **Objectively Evaluate Processes**
        i. Boost employee participation by promoting an environment where reporting of progress is encouraged. This will aid in the identifying process.
        ii. Establish clearly stated rules for the evaluation of processes against the process descriptions, standards, and procedures.
    b. **Objectively Evaluate Work Products**
        i. Establish work products to be evaluated.
        ii. Evaluate work products before they are delivered to the customer and during milestones against the process descriptions, standards, and procedures.

2. **Provide Objective Insight**

   a. **Communicate and Resolve Noncompliance Issues**

      i. Communicate and resolve each noncompliance issue with the team members.

      ii. Document noncompliance issues when they cannot be resolved within the project.

      iii. Resolve the noncompliance by changing the process descriptions, standards, or procedures.

      iv. Report the noncompliance issues that cannot be resolved to the appropriate level of management and act on noncompliance issues.

   b. **Establish Records**

      i. Establish and maintain records of the quality assurance activities such as evaluation logs and quality assurance reports.

      ii. Document the process and product quality assurance activities in detail such that status and results are known.

2.6. **Software Configuration Management**

2.6.1. **Goal**

The goal of Configuration Management process is to establish and maintain consistency of work products using configuration identification, configuration control and configuration audits throughout its lifecycle.

2.6.2. **Key Practices**

1. **Establish Baselines**

   a. **Identify Configuration Items**

      i. Source code
      ii. System Requirements Specification
      iii. Project plan
      iv. Use cases
      v. Test data
      vi. Designs

b. **Establish a Configuration Management System**

   i. Store, update and retrieve configuration items in a configuration management system.

   ii. Create configuration management reports from the configuration management system for traceability and auditing.

   iii. Archive and backup configuration management items.

   iv. Revise the configuration management items when needed.

c. **Create or Release Baselines**

   i. Create or release baselines from configuration items.

   ii. Document the set of configuration items that are contained in a baseline.

1. **Track and Control Changes**

   a. **Track Change Requests**

      i. If there are any change requests, record the changes in the change request database.

      ii. Analyze the impact of changes to ensure that the new changes are consistent with the project requirements.

   b. **Control Configuration Items**

      i. Obtain appropriate authorization before changing configuration items to maintain the correctness and integrity of the configuration items.

2. **Establish Integrity**

   a. **Establish Configuration Management Records**

      i. Record configuration management actions in detail in the revision history of configuration items, status and change log.

   b. **Perform Configuration Audits**

      i. Verify that the configuration management records correctly identify the configuration items.

      ii. Rectify discrepancies if needed.