



## CMMI Level 2 Definition

RoomieLah: Finding university roommates made easy!

**Arora Srishti** – Project Manager, Back-end Developer

**Pandey Pratyush** – Lead Developer

**Agarwal Gopal** - Release Engineer

**Surawar Sanath Sachin** – Front-end Developer

**Iyer Rajagopal** – Back-end Developer

**Acharya Atul** – QA Manager

**Tayal Aks** – QA Engineer

Team Strawhats  
School of Computer Science & Engineering  
Nanyang Technological University, Singapore

Submitted to—  
School of Computer Science & Engineering  
Nanyang Technological University, Singapore

# Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Atul Acharya	06/04/2022	Srishti Arora	07/04/2022	Initial Release Management Plan
1.2	Sanath Surawar	09/04/2022	Srishti Arora	10/04/2022	CMMI Checklist
1.3	Sanath Surawar	10/04/2022	Srishti Arora	10/04/2022	Level 2 KPAs, Generic Goals and Practices
1.4	Atul Acharya	11/04/2022	Srishti Arora	12/04/2022	Specific Goals and Practices

# TABLE OF CONTENTS

1	EXECUTIVE SUMMARY	5
	1.1 Purpose	5
	1.2 Summary of Definition	5
2	DESCRIPTION	6
3	LEVEL 2 KPAs	7
	3.1 Requirement Management (REQM)	7
	3.2 Project Planning (PP)	8
	3.3 Project Monitoring and Control (PMC)	8
	3.4 Process and Product Quality Assurance (PPQA)	9
	3.5 Configuration Management (CM)	9
	3.6 Measurement and Analysis (MA)	10
	3.7 Supplier Agreement Management (SAM)	10
4	GENERIC GOALS AND PRACTICES	10
	4.1 Commitment to perform	10
	4.2 Ability to perform	11
	4.3 Activities to perform	11
	4.4 Measurement	11
	4.5 Verification of Implementation	12
5	SPECIFIC GOALS AND PRACTICES	12
	5.1 Process Management	12

	5.2 Project Management	14
	5.3 Engineering	15
6	APPROVALS	17
7	CMMI AUDIT CHECKLIST	18
8	CMMI INTERVIEW AFFIRMATION QUESTIONS	24

# 1. EXECUTIVE SUMMARY

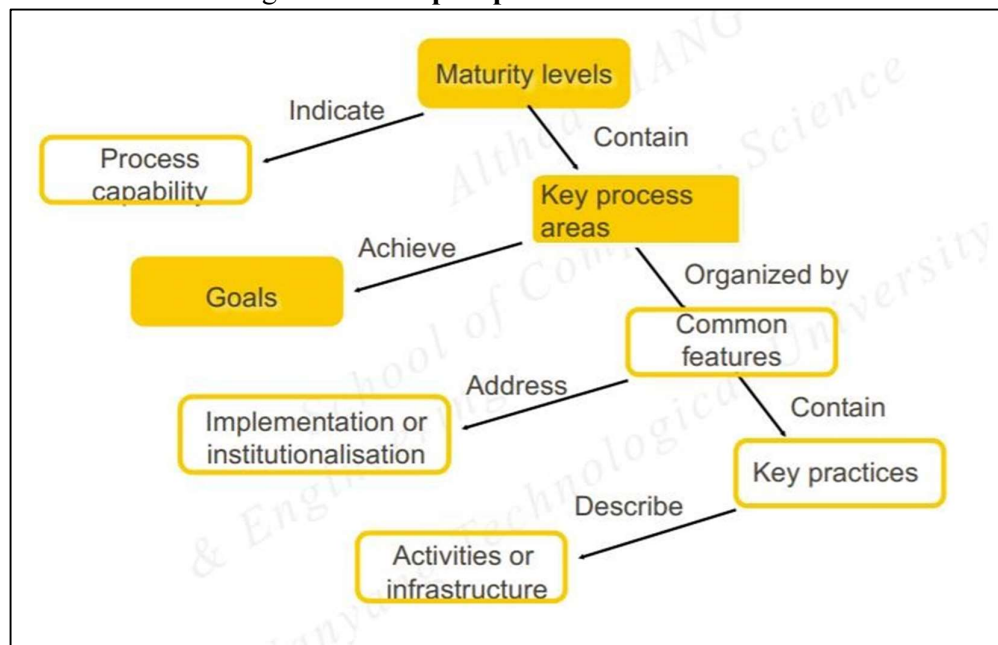
## 1.1 Purpose

This document's purpose is to establish appropriate baselines and guidelines to be used by Team StrawHats for further improvement of RoomieLah as an application and across Team StrawHats as a whole. It provides information on how to transition into a higher level of the Capability Maturity Model Integration model. The coalescence of all the necessary information to transition to a higher stage in the hierarchy of levels defined by this model, helps serve as a consistent standard of reference for all members of Team StrawHats.

## 1.2 Summary of definition

These are the key characteristics Team Strawhats wishes to emulate once it initiates operating on CMMI Level 2:

- Work carried out according to **planned** processes
- **Clearly** defined roles and responsibilities
- **Quantitative basis** for judging quality and analysis problems
- Performance measures **generally achieved**
- **Monitoring** of quality of software products and customer satisfaction by management
- Schedules and budgets based on **past performance**

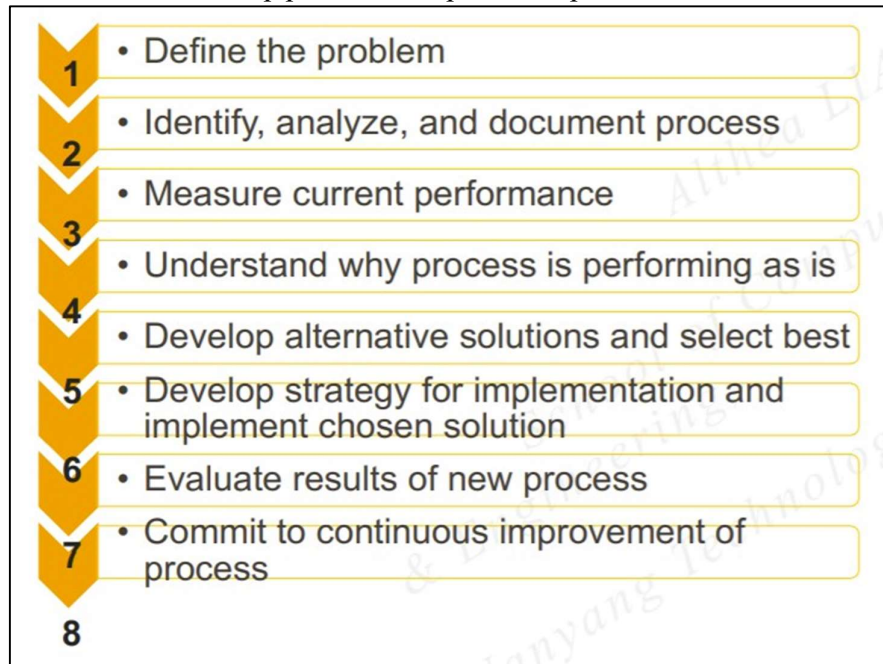


**Figure 1:** Flowchart used to determine the CMMI level of Team Strawhats

## 2.DESCRPTION

The Capability Maturity Model Integration (CMMI) is a process and behavioral model that helps organizations streamline process improvement and encourage productive, efficient behaviors that decrease risks in software, product, and service development.

At Team StrawHats, we used the CMMI model to reflect on the way the development of Strawhats took place and use the CMMI standards to further improve our development process. We followed a detailed, 8 – step process to improve our processes:



**Figure 2:** *Eight step Process Improvement Model* (Source – CZ3002 – Dr Quah TS, Jon)

The above model helps to flag and remove issues detected in earlier versions of the software development process. It includes a number of steps covering a broad spectrum of work: from definition of the problem to commitment from team members for continual improvement. It serves as an effective guide in outlining steps needed to go to the next level of maturity in the CMMI model.

The five maturity models defined by CMMI are: *Initial*, *Managed*, *Defined*, *Quantitatively Managed*, and *Optimizing*. Team Strawhats is presently at the *Initial* stage. Hence, the purpose of the document is to provide appropriate guidance in moving on to the next stage, that is, the *Managed* maturity level.

## 3. LEVEL 2 KPAs

### 3.1 Requirement Management

KPA stands for Key Process Areas. A Process Area is a cluster of related practices in an area that, when implemented collectively, satisfy a set of goals considered important for making significant improvement in that area. All CMMI process areas are common to both continuous and staged representations. The CMMI process areas can be grouped into the following categories, to understand the links between them:

- Project Management
- Process Management
- Engineering
- Support

A process area is satisfied when company processes cover all the generic and specific goals and practices **for that process area**.

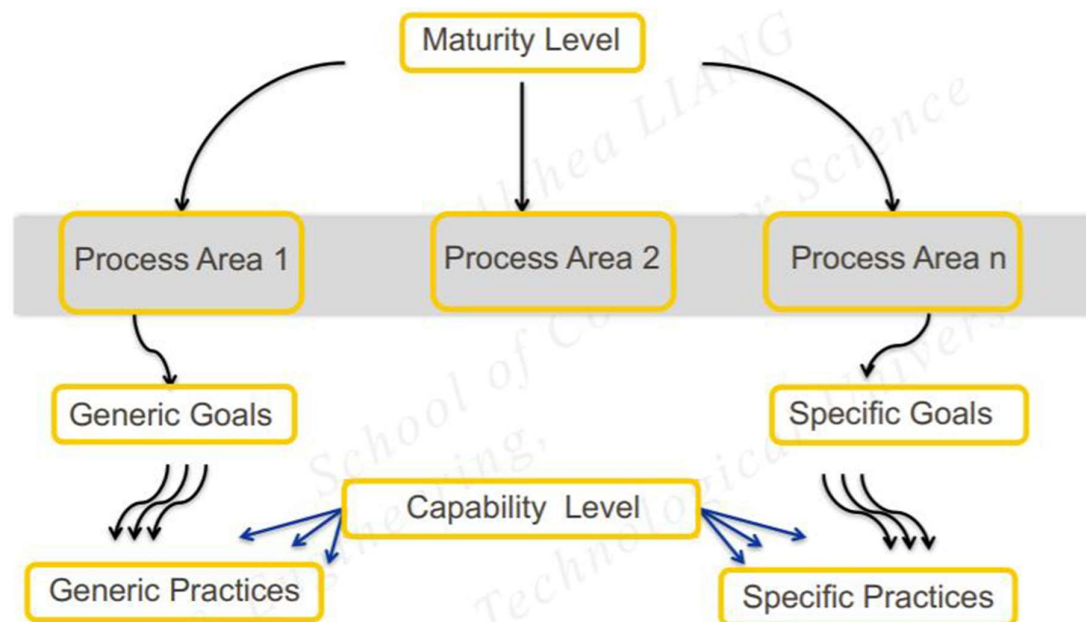
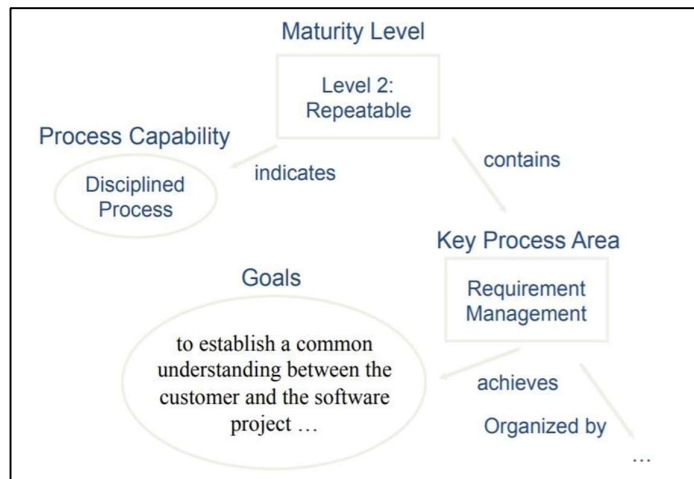


Figure 3: Definition of CMMI 2 (Source – CZ3002 Lecture Notes – Dr Quah TS, Jon)

From Figure 3, the importance of Key Process Areas can be clearly seen, as they form the basis of determining the maturity level and subsequently deciding the generic goals and processes, as well as the specific goals and processes, based on the organizational maturity level.

Each process area drives the generic goals, which in turn drive the generic practices (together with the capability level), needed to ensure higher maturity of the processes. Hence, we at Team Strawhats aim to utilize the Key Process Areas to identify areas of improvement.

Below, we define the various level 2 KPAs in CMMI:



**Figure 4:** Requirement Management - Level 2 KPA in CMMI

The purpose of Requirement Management is managing the requirements of several components within a software project. Further, REQM aims to align and maintain a high degree of consistency between the specified requirements and the project's actual plan, as well as the work products.

The detailed purpose of REQM is as follows:

- to establish a common understanding between the customer and the requirements that the customer wishes to be addressed by the project.
- serves as the basis for further planning and managing RoomieLah as a software project.

### 3.2 Project Planning (PP)

Project Planning (PP) is an important aspect that governs the success of any project, both in terms of customer satisfaction as well as the quality. Through appropriate, verified and proof-read plans, the team can ensure that the project meets both time and budget constraints. Project Planning for RoomieLah would help achieve the following key objectives:

- establishment of plans that are reasonable for managing RoomieLah
- providing an effective baseline as well as the boundaries within which the work for the software project should be performed.

### 3.3 Project Monitoring and Control (PMC)

Project Monitoring and Control (PMC) has great weightage in ensuring the completion of the project, in terms of the end product and its quality. It is closely integrated with



the *Development* phase of the project lifecycle, which includes the entire *Software Development Lifecycle* (SDLC).

Monitoring and control procedures demand a high degree of senior management sponsorship to ensure proper alignment of expectations between the expected and actual procedures and processes involved in the development of the software project. It also includes **tracking and oversight** to provide increased visibility into the actual progress, to enable the management to take effective actions when the performance and quality of the software project deviates significantly from the expected performance defined in prior software plans.

### 3.4 Process and Product Quality Assurance (PPQA)

Process and Product Quality Assurance is needed to ensure consistent high quality of RoomieLah as a software. This in turn helps to ensure an appropriate level of portability and maintainability of the software. PPQA aims to provide senior staff and management, an objective insight into the Quality Assurance procedures and processes, the level of adoption and the degree to which they reflect, in the product as well as the processes.

The main aims of PPQA can be summarised as:

- High visibility and transparency into the processes being used by the software project and of the products being built, to intervene and perform timely quality control if deemed absolutely necessary.
- Ensure absolute compliance with applicable organizational standards and procedures, by reviewing and auditing the software products and activities being performed.

### 3.5 Configuration Management (CM)

Software Configuration Management (CM) can be considered as one of the most important steps of the project lifecycle, as it is the part that is most closely integrated with customer usage of the software.

The purpose of the Software Configuration Management is as follows:

- establish and maintain integrity of various products under the overall software project, throughout the life cycle of the software
- helps in determining and checkpointing the configuration of the software at different points in time, that is, how the software has changed over a certain interval of time. It helps in systematically controlling changes to the configuration of the software project and maintaining a high level of traceability of the software. This in turn helps in increasing the maintainability of the software.

### **3.6 Measurement and Analysis (MA)**

The purpose of Measurement and Analysis (MA) is to develop and sustain capability of measurement of various aspects of software projects (schedule estimation, budget estimation etc.) to fulfil information needs of senior management.

MA is an important supplement to Project Monitoring and Control as it helps in making informed decisions such as changes needed if overshooting time / budget etc, tools and equipment needed to perform certain tasks etc. However, MA is also an independent KPA in itself that it takes place continuously throughout the project lifecycle, from planning all the way to completion, and beyond that, when software is maintained, and Change Requests (CRs) start being initiated. Analysis of the measurements performed should be accurate to make the best decision about the next steps involved.

### **3.7 Supplier Agreement Management (SAM)**

Supplier Agreement Management (SAM) is important in that it is absolutely essential in building the software and making it available to the general user base. It is intended to manage the acquisition of supplies from the suppliers, which covers a wide variety of things. The acquisition of these tools are necessary in making the product possible in the first place; hence it is important to have a separate well – defined process for this.

SAM assumes that a formal agreement with the suppliers already exists and is hence responsible for covering the aspects that occur post that. There are several factors involved in the agreements with the supplier, such as business alliances, process capabilities, technical considerations etc. It is important to select suppliers based on the ability to perform.

## **4. GENERIC GOALS AND PRACTICES**

### **4.1 Commitment to perform**

Commitment to perform represents the actions that an organization must take to ensure the processes and procedures established will be followed with the appropriate performance standards and will continue to endure. This commitment must be a guarantee from every individual within the organization, that their performance shall be of the highest standards, in accordance with established organization policies and quality.

The commitment to perform is an essential aspect of the overall software project lifecycle, as a lapse from the team members can signal a waning team effort and consequently, a poor quality product. We at Team Strawhats adopted the following general practices to ensure the highest commitment to perform from every team member:

- Increased senior management sponsorship, that is, individual briefings and reminder of high-quality work, by the Project Manager and Lead Developer at RoomieLah.
- Prior specification and establishment of team-wide policies and quality of procedures and individual work.
- Regular inspections by Quality Assurance Manager to gauge the individual commitment level and its corresponding reflection in performance.

#### **4.2 Ability to perform**

The ability to perform in the project is highly correlated to the work environment that exists when the team is working together. It is imperative to effectuate an environment where each individual team member can perform to their best. To enable this, the organization must take similar efforts on their end to provide the resources to enable them to do so. Along those lines, Team Strawhats followed the following practices to increase overall ability of the team:

- Regular management reviews, with special focus on non-technical aspects of the project, namely time constraints, availability of resources, like full cloud access etc.
- Frequent checks with team members to collate resources needed to complete at least the work for the next deliverable.
- An anonymous peer review platform to voice out any discrepancies within the team.

#### **4.3 Activities to perform**

These include and describe the roles for different individuals as well as the activities that each needs to execute, to achieve the different Key Process Areas (KPAs) defined earlier.

Team Strawhats adopted the following practices to track the activities to perform:

- Validation of Project Plan and documents outlining work of different team members
- Daily stand-ups, to track activities performed and assign new roles if required

#### **4.4 Measurement**

To ensure timely completion of a project and to keep it within limits of the budget, it is important to adopt methodologies that can appropriately measure the software, in terms of the schedule (time for various components in the project lifecycle) and the budget.

To this effect, Team Strawhats had the following general practices, for accurate measurement:

- Function Point model for time effort and duration estimation
- Cross-checking with the latest **COCOMO II** model to verify the results of the Function Point model and to make more detailed and accurate estimations using the adjusted function points and Caper Jones' statistics for lines of code.

## 4.5 Verification of Implementation

Verification and Validation (V&V) is an important process that occurs parallel to the Software Development Lifecycle (SDLC). In verification, the organization checks whether the processes, product and procedures meet the stated and established policies and standards. In validation, it is confirmed whether the product meets all the client requirements, to their satisfaction or not.

Verification and Validation is taken extremely seriously by Team Strawhats. Hence, we adopted:

- A dedicated quality assurance team, with a Quality Assurance Manager and a Quality Assurance Engineer who work to establish a **quality culture** within Team Strawhats
- Regular verification checks of software development process, by quality engineers and senior management, to prevent deviation of established levels of high quality

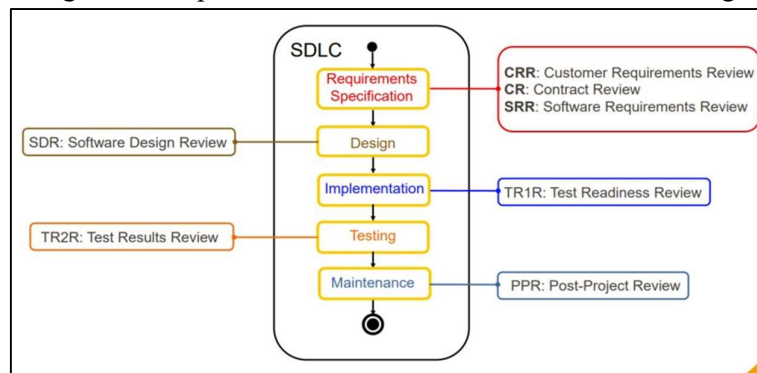


Figure 5: Reviews by Team Strawhats, at SDLC Milestones

# 5. SPECIFIC GOALS AND PRACTICES

## 5.1 Process Management

### 5.1.1 Organisation Process Focus

#### Specific Goal

1. To Plan and Implement Process Action.
2. Deploy Organization Process Assets and Incorporate Experiences.

#### Specific Practices

- 1.1 Identify strategies, approaches, and actions to address identified process improvements.
- 1.2 Establish process action teams to implement actions.
- 1.3 Negotiate and document commitments among process action teams and revise their process action plans, as necessary.

- 1.4 Conduct joint reviews with process action teams and relevant stakeholders to monitor the progress and results of process actions.
- 2.1 Deploy organizational process assets across the organization.
- 2.2 Document changes to organizational process assets.
- 2.3 Deploy changes that were made to organizational process assets across the organization.
- 2.4 Provide guidance and consultation on the use of organizational process assets.

## **5.1.2 Organisational Process Definition**

### Specific Goal

- 1. To Establish Organizational Process Assets.

### Specific Practices

- 1.1 Decompose each standard process into constituent process elements to the detail needed to understand and describe the process.
- 1.2 Specify the critical attributes of each process element.
- 1.3 Specify relationships among process elements.
- 1.4 Ensure that the organization's set of standard processes adheres to applicable policies, standards, and models.

## **5.1.3 Organisation Training**

### Specific Goal

- 1. Establish an Organizational Training Capability.
- 2. Provide Training.

### Specific Practices

- 1.1 Analyse the organization's strategic business objectives and process improvement plan to identify potential training needs.
- 1.2 Document the strategic training needs of the organization.
- 1.3 Determine the roles and skills needed to perform the organization's set of standard processes.
- 1.4 Document the training needed to maintain the safe, secure, and continued operation of the business.
- 2.1 Select those who will receive the training necessary to perform their roles effectively.
- 2.2 Schedule the training, including any resources, as necessary (e.g., facilities, instructors).
- 2.3 Deliver the training.

2.4 Track the delivery of training against the plan.

## **5.2 Project Management**

### **5.2.1 Project Planning**

#### Specific Goal

1. To Establish Estimates.
2. Develop a Project Plan.

#### Specific Practices

- 2.1 Develop a WBS.
- 2.2 Define the work packages in sufficient detail so that estimates of project tasks, responsibilities, and schedule can be specified.
- 2.3 Identify products and product components to be externally acquired.
- 2.4 Identify work products to be reused.
- 2.1 Identify major milestones.
- 2.2 Identify schedule assumptions.
- 2.3 Identify constraints.
- 2.4 Identify task dependencies.
- 2.5 Establish and maintain the budget and schedule.

### **5.2.2 Supplier Agreement Management (SAM)**

#### Specific Goal

1. To Establish Supplier Agreements.
2. Satisfy Supplier Agreements.

#### Specific Practices

- 1.1 Determine type of acquisition.
- 1.2 Establish and document criteria for evaluating potential suppliers.
- 1.3 Identify potential suppliers and distribute solicitation material and requirements to them.
- 1.4 Evaluate proposals according to evaluation criteria.
- 2.1 Monitor supplier progress and performance (e.g., schedule, effort, cost, technical performance) as defined in the supplier agreement.
- 2.2 Conduct reviews with the supplier as specified in the supplier agreement.
- 2.3 Use the results of reviews to improve the supplier's performance and to establish and nurture long-term relationships with preferred suppliers.
- 2.4 Monitor risks involving the supplier and take corrective action, as necessary.

## 5.3 Engineering

### 5.3.1 Requirements Development (RD)

#### Specific Goal

1. Develop Customer Requirements.
2. Develop Product Requirements.
3. Analyse and Validate Requirements.

#### Specific Practices

- 1.1 Engage relevant stakeholders using methods for eliciting needs, expectations, constraints, and external interfaces.
- 1.2 Translate stakeholder needs, expectations, constraints, and interfaces into documented customer requirements.
- 1.3 Establish and maintain a prioritization of customer functional and quality attribute requirements.
- 1.4 Define constraints for verification and validation.
- 2.1 Develop requirements in technical terms necessary for product and product component design.
- 2.2 Derive requirements that result from design decisions.
- 2.3 Develop architectural requirements capturing critical quality attributes and quality attribute measures necessary for establishing the product architecture and design.
- 2.4 Establish and maintain relationships between requirements for consideration during change management and requirements allocation.
- 2.5 Identify interfaces both external to the product and internal to the product (e.g., between functional partitions or objects).
- 2.6 Develop the requirements for the identified interfaces.
- 3.1 Develop operational concepts and scenarios that include operations, installation, development, maintenance, support, and disposal as appropriate.
- 3.2 Define the environment in which the product or product component will operate, including boundaries and constraints.
- 3.3 Review operational concepts and scenarios to refine and discover requirements.
- 3.4 Develop a detailed operational concept, as products and product components are selected, that defines the interaction of the product, the end user, and the environment, and that satisfies the operational, maintenance, support, and disposal needs.

### 5.2.3 Technical Solution (TS)

#### Specific Goal

1. To Select Product Component Solutions.
2. Develop the Design.
3. Implement the Product Design

#### Specific Practices

- 1.1 Identify screening criteria to select a set of alternative solutions for consideration.
- 1.2 Identify technologies currently in use and new product technologies for competitive advantage.
- 1.3 Identify candidate COTS products that satisfy the requirements.
- 1.4 Establish and maintain the documentation of the solutions, evaluations, and rationale.
- 2.1 Establish and maintain criteria against which the design can be evaluated.
- 2.2 Identify, develop, or acquire the design methods appropriate for the product.
- 2.3 Ensure that the design adheres to applicable design standards and criteria.
- 2.4 Ensure that the design adheres to allocated requirements.
- 2.5 Document the design.
- 3.1 Use effective methods to implement the product components.
- 3.2 Adhere to applicable standards and criteria.
- 3.3 Conduct peer reviews of the selected product components.
- 3.4 Perform unit testing of the product component as appropriate.
- 3.5 Revise the product component, as necessary.

### **5.3.3. Product Integration (PI)**

#### Specific Goal

1. Prepare for Product Integration.
2. Ensure Interface Compatibility.
3. Assemble Product Components and Deliver the Product.




#### Specific Practices

- 1.1 Identify the product components to be integrated.
- 1.2 Identify the verifications to be performed during the integration of the product components.
- 1.3 Periodically review the product integration strategy and revise as needed.
- 1.4 Record the rationale for decisions made and deferred.
- 1.5 Maintain the product integration environment throughout the project.
- 2.1 Review interface data for completeness and ensure complete coverage of all interfaces.
- 2.2 Ensure that product components and interfaces are marked to ensure easy and correct connection to the joining product component.



- 2.3 Periodically review the adequacy of interface descriptions.
- 2.4 Maintain a repository for interface data accessible to project participants.
- 3.1 Track the status of all product components as soon as they become available for integration.
- 3.2 Ensure that product components are delivered to the product integration environment in accordance with the product integration strategy and procedures.
- 3.3 Check the configuration status against the expected configuration.
- 3.4 Perform a pre-check (e.g., by a visual inspection, using basic measures) of all the physical interfaces before connecting product components together.

## 6. APPROVALS

Signature:		Date:	11/04/2022
Print Name:	Srishti Arora		
Title:	Mrs		
Role:	Product Manager		
Signature:		Date:	11/04/2022
Print Name:	Atul Acharya		
Title:	Mr		
Role:	Quality Assurance Manager		
Signature:		Date:	11/04/2022
Print Name:	Aks Tayal		
Title:	Mr		
Role:	Quality Assurance Engineer		

## 7. CMMI AUDIT CHECKLIST

KPAS	Specific Goal	Specific Practice	Tick
Organisation Process Focus	To Plan and Implement Process Action	Identify strategies, approaches, and actions to address identified process improvements	✓
		Establish process action teams to implement actions	✓
		Negotiate and document commitments among process action teams and revise their process action plans, as necessary	✓
		Conduct joint reviews with process action teams and relevant stakeholders to monitor the progress and results of process actions	✓
	Deploy Organization Process Assets and Incorporate Experiences	Deploy organizational process assets across the organization	✓
		Document changes to organizational process assets	✓
		Deploy changes that were made to organizational process assets across the organization	✓
		Provide guidance and consultation on the use of organizational process assets	✓
Organisational Process Definition	To Establish Organizational Process Assets	Decompose each standard process into constituent process elements to the detail needed to understand and describe the process	✓
		Specify the critical attributes of each process element	✓
		Specify relationships among process elements	✓

		Ensure that the organization's set of standard processes adheres to applicable policies, standards, and models	✓
Organisation Training	Establish an Organizational	Analyse the organization's strategic business objectives and process	✓
	Training Capability	improvement plan to identify potential training needs	
		Document the strategic training needs of the organization	✓
		Determine the roles and skills needed to perform the organization's set of standard processes	✓
		Document the training needed to maintain the safe, secure, and continued operation of the business	✓
	Provide Training	Select those who will receive the training necessary to perform their roles effectively	✓
		Schedule the training, including any resources, as necessary (e.g., facilities, instructors)	✓
		Deliver the training	✓
		Track the delivery of training against the plan	✓
Project Planning	To Establish Estimates	Develop a WBS	✓
		Define the work packages in sufficient detail so that estimates of project tasks, responsibilities, and schedule can be specified	✓
		Identify products and product components to be externally acquired	✓
		Identify products and product components to be externally acquired	✓
		Identify work products to be reused	✓
	Develop a Project Plan	Identify major milestones	✓
		Identify schedule assumptions	✓

		Identify constraints	✓
		Identify task dependencies	✓
		Establish and maintain the budget and schedule	✓
		Establish corrective action criteria	✓
Supplier Agreement Management (SAM)	To Establish Supplier Agreements	Determine type of acquisition	✓
		Establish and document criteria for evaluating potential suppliers	✓
		Identify potential suppliers and distribute solicitation material and requirements to them	✓
		Evaluate proposals according to evaluation criteria	✓
	Satisfy Supplier Agreements	Monitor supplier progress and performance (e.g., schedule, effort, cost, technical performance) as defined in the supplier agreement	✓
		Conduct reviews with the supplier as specified in the supplier agreement	✓
		Use the results of reviews to improve the supplier's performance and to establish and nurture long-term relationships with preferred suppliers	✓
Requirements Development (RD)	Develop Customer Requirements	Engage relevant stakeholders using methods for eliciting needs, expectations, constraints, and external interfaces	✓
		Translate stakeholder needs, expectations, constraints, and interfaces into documented customer requirements	✓
		Establish and maintain a prioritization of customer functional and quality attribute requirements	✓

		Define constraints for verification and validation	✓
	Develop Product Requirements	Develop requirements in technical terms necessary for product and product component design	✓
		Derive requirements that result from design decisions	✓
		Develop architectural requirements capturing critical quality attributes and quality attribute measures necessary for establishing the product architecture and design	✓
		Establish and maintain relationships between requirements for consideration during change management and requirements allocation	✓
		Identify interfaces both external to the product and internal to the product (e.g., between functional partitions or objects)	✓
		Develop the requirements for the identified interfaces	✓
	Analyse and Validate Requirements	Develop operational concepts and scenarios that include operations, installation, development, maintenance, support, and disposal as appropriate	✓
		Define the environment in which the product or product component will operate, including boundaries and constraints	✓
		Review operational concepts and scenarios to refine and discover requirements	✓
		Develop a detailed operational concept, as products and product components are selected, that defines the interaction of the product, the end user, and the environment, and that satisfies the operational, maintenance, support, and disposal needs	✓

Technical Solution (TS)	To Select Product Component Solutions	Identify screening criteria to select a set of alternative solutions for consideration	✓
		Identify technologies currently in use and new product technologies for competitive advantage	✓
		Identify candidate COTS products that satisfy the requirements	✓
		Establish and maintain the documentation of the solutions, evaluations, and rationale	✓
	Develop the Design	Establish and maintain criteria against which the design can be evaluated	✓
		Identify, develop, or acquire the design methods appropriate for the product	✓
		Ensure that the design adheres to applicable design standards and criteria	✓
		Ensure that the design adheres to allocated requirements	✓
		Document the design	✓
	Implement the Product Design	Use effective methods to implement the product components	✓
		Adhere to applicable standards and criteria	✓
		Conduct peer reviews of the selected product components	✓
		Perform unit testing of the product component as appropriate	✓
		Revise the product component, as necessary	✓
	Product Integration (PI)	Identify the product components to be integrated	✓
		Identify the verifications to be performed during the integration of the product components	✓
		Periodically review the product integration strategy and revise as needed	✓

		Record the rationale for decisions made and deferred	✓
		Maintain the product integration environment throughout the project	✓
	Ensure Interface Compatibility	Review interface data for completeness and ensure complete coverage of all interfaces.	✓
		Ensure that product components and interfaces are marked to ensure easy and correct connection to the joining product component.	✓
		Periodically review the adequacy of interface descriptions	✓
		Maintain a repository for interface data accessible to project participants.	✓
	Assemble Product Components and	Track the status of all product components as soon as they become available for integration.	✓
	Deliver the Product	Ensure that product components are delivered to the product integration environment in accordance with the product integration strategy and procedures.	✓
		Check the configuration status against the expected configuration	✓
		Perform a pre-check (e.g., by a visual inspection, using basic measures) of all the physical interfaces before connecting product components together	✓

## **8. CMMI INTERVIEW AFFIRMATION QUESTIONS**

1. Where and how is the framework to plan, organize and control the work done on the project documented? Do you have a WBS? Where is it documented?
2. What parameters do you use to estimate work products and tasks (document size, critical computer resources, and source size, bandwidth)?
3. How do you estimate the project's effort and costs? What rationale is used?
4. How do you determine the project's life cycle phases and where is it documented for reference?