



# **BACHELOR OF COMPUTER APPLICATIONS**

## **SEMESTER 5**

**DCA3103**

**SOFTWARE ENGINEERING**

# Unit 6

## Assessment of Process Life-cycle Models

### Table of Contents

SL No	Topic	Fig No / Table / Graph	SAQ / Activity	Page No
1	<a href="#">Introduction</a>	-	-	3 - 4
1.1	<a href="#">Learning Objectives</a>	-	-	
2	<a href="#">Overview of the Assessment of the Process</a>	-	-	5 - 10
2.1	<a href="#">Approaches to software process assessment</a>	-	-	
3	<a href="#">Dimensions of Process</a>	-	-	11 - 12
4	<a href="#">The Need for a business model in Software Engineering.</a>	-	-	13 - 14
5	<a href="#">Summary</a>	-	-	15
6	<a href="#">Self-assessment Questions</a>	-	-	16 - 17
7	<a href="#">Self-assessment Answers</a>	-	-	18
8	<a href="#">Terminal Questions</a>	-	-	18
9	<a href="#">Terminal Answers</a>	-	-	19

## 1. INTRODUCTION

Process life-cycle models are helpful frameworks used in a variety of industries to direct and manage process development, implementation, and maintenance. These models offer a methodical way to guarantee consistency, quality, and effectiveness over a process' entire life cycle, from its inception to retirement. Organizations may improve process outcomes, promote stakeholder collaboration, and effectively manage risk by following a clearly defined set of stages and activities.

The purpose of assessing process life-cycle models is to determine their suitability and effectiveness in a specific project or organizational context. It involves evaluating various aspects of the model to ensure it aligns with the nature and complexity of the process being developed. This assessment helps organizations make informed decisions about selecting the most appropriate life-cycle model, thereby increasing the chances of successful process implementation and maintenance.

Simply having a software process in place does not guarantee that software will be created on time, that it will meet the needs of the client, or that it will exhibit the technical features that will result in long-term quality characteristics.

Process patterns need to be used in combination with good software engineering techniques.

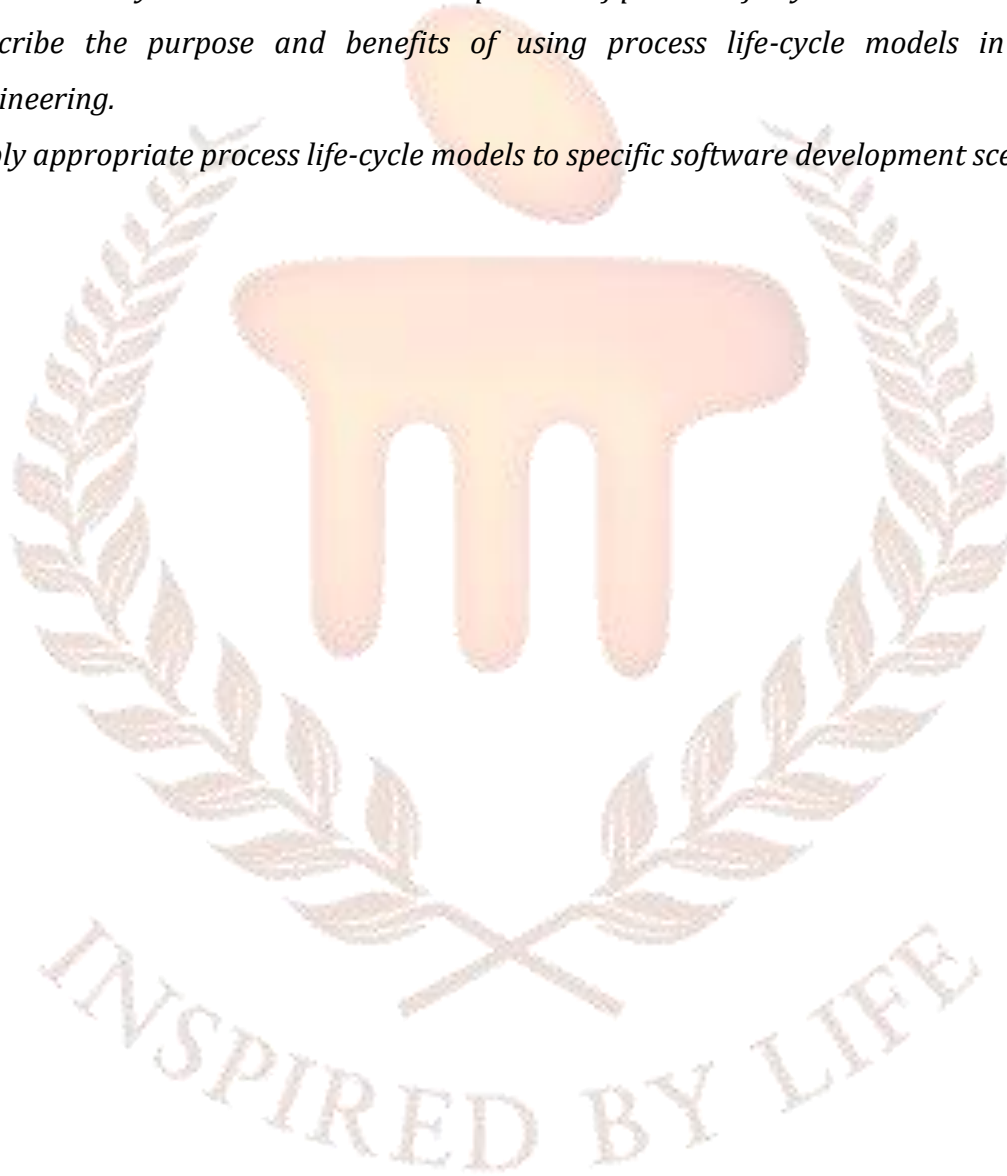
Additionally, the process can be evaluated to make sure that it meets several fundamental process requirements that have been demonstrated to be crucial for successful software engineering.

Over the past few decades, a wide range of methods like SCAMPI, CBA IPI, ISO/IEC15504, and ISO 9001:2000 for software process evaluation and improvement have been proposed.

## 1.1 Learning Objectives:

*After completion of each unit, you will be able to:*

- ❖ *Recall the various process life-cycle models commonly used in software development.*
- ❖ *Explain the key characteristics and components of process life-cycle models.*
- ❖ *Describe the purpose and benefits of using process life-cycle models in software engineering.*
- ❖ *Apply appropriate process life-cycle models to specific software development scenarios.*



## 2. OVERVIEW OF THE ASSESSMENT OF THE PROCESS

Software engineering process assessment is assessing and analysing the procedures followed by a company to create software products. It tries to evaluate the development, effectiveness, and efficiency of these processes and identify potential areas of development. The evaluation provides useful details about the company's capacity to consistently generate high-quality software and deliver it on time and under budget.

**An outline of the software engineering assessment procedure is given below:**

### **1. Assessment Planning / Evaluation Planning:**

- Define the assessment's parameters and goals.
- Decide on the evaluation standards, such as industry norms, best practices, or process frameworks.
- Determine the assessment team, which may consist of internal or external assessors with expertise in software engineering processes.

### **2. Data Gathering:**

- Gather information and proof about the software engineering procedures that are being evaluated.
- This could involve reviewing documents, speaking with stakeholders in person, watching the process in action, and looking over deliverables and artefacts.

### **3. Process analysis:**

- Analyse the information gathered to assess how the software engineering processes are doing right now.
- Identify any gaps or potential areas for development.
- Evaluate the process maturity.

### **4. Findings and Recommendations:**

- Record the assessment results, highlighting the strengths, flaws, and potential improvement areas.
- Give suggestions for process improvements based on the gaps and best practices you've found.
- Sort the suggestions in order of importance and feasibility for execution.



**5. Action Planning:**

- Create a plan of action to address the areas that need improvement.
- Establish clear objectives, roles, and deadlines for carrying out the recommended changes.
- Set up measures to gauge the success of the improvement activities and allocate resources accordingly.

**6. Implementation and Monitoring:**

- Execute the action plan by implementing the identified process improvements.
- Monitor the progress of the improvement initiatives, track the defined metrics, and assess the impact of the changes on software development processes.

**7. Follow-Up Assessment:**

- To measure the success of the implemented modifications, conduct follow-up assessments.
- To evaluate progress, compare the processes' present state to the results of the initial assessment.
- Find any other holes or locations that require fixing.

**2.1 Approaches To Software Process Assessment**

Several different approaches to software process assessment and improvement have been mentioned below:

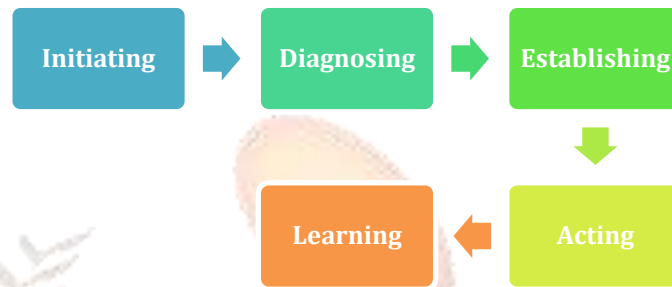
**1. Standard CMMI Assessment Method for Process Improvement (SCAMPI):**

The Standard CMMI Assessment Method for Process Improvement (SCAMPI) is a structured methodology that gives organisations a way to evaluate and enhance their processes using the Capability Maturity Model Integration (CMMI) framework.

SCAMPI incorporates a five-step process assessment model, which consists of the following phases:

- i. Initiating
- ii. Diagnosing
- iii. Establishing

- iv. Acting and
- v. Learning.



*i. Initiating:*

During this phase, the organisation establishes the assessment's scope, objectives, and criteria. The assessment team is assembled, roles and responsibilities are assigned, and the assessment plan is developed.

*ii. Diagnosing:*

Data gathering and analysis are done during the diagnostic phase in order to compare the organization's processes to the CMMI model.

In order to determine the process maturity and capability, the assessment team conducts interviews, examines objects, and compiles data.

The objective is to identify the processes of the organization's strengths, flaws and potential improvement areas.

*iii. Establishing:*

During establishing phase, the evaluation results are examined and recommendations for process improvement are developed.

The evaluation team collaborates with the organisation to prioritise improvement actions, create a roadmap, and set improvement objectives.

In order to implement the suggested improvements, the organisation sets up the appropriate infrastructure, resources, and strategies.

iv. *Acting:*

The acting phase focuses on implementing the process improvements identified in the previous phase. The organisation executes the planned improvement actions, tracks development, and evaluates the success of the improvements put in place.

Executing process improvement initiatives, training staff, updating documentation and making sure the changes are incorporated into the organization's practises are all part of this phase.

v. *Learning:*

The value of continuous learning and improvement is emphasised during the learning phase.

The company gathers feedback, assesses the effects of the applied changes, and measures the efficiency of the modifications.

Lessons learnt from the evaluation and improvement initiatives are recorded and shared in order to expand the organization's knowledge base and support continuous progress.

## **2. CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**

The idea behind CMM-Based Appraisal for Internal Process Improvement (CBA IPI) is to evaluate the relative maturity of a software organization's processes using the Capability Maturity Model (CMM) created by the Software Engineering Institute (SEI).

To assess the organization's current level of process maturity, identify strengths and flaws and offer direction for internal process improvement initiatives, CBA IPI is used as a diagnostic approach.

The following are the most important concepts to comprehend regarding CMM-Based Appraisal for Internal Process Improvement (CBA IPI):





- i. *Purpose:* The SEI CMM is used as a reference model in CBA IPI to determine the relative maturity of software organisations.  
The main objective is to determine the organization's software development processes' strengths and weaknesses to guide internal process improvement initiatives.
- ii. *Diagnostic Technique:* The organization's processes are evaluated by CBA IPI in comparison to the CMM using a diagnostic technique. It involves evaluating the organization's procedures, objects, and performance data to establish the level of maturity attained in the various CMM-specified process domains.
- iii. *Relative Maturity Assessment:* The five maturity levels specified in the CMM—Initial, Managed, Defined, Quantitatively Managed, and Optimizing—are compared to determine the organization's relative maturity levels. This evaluation is provided by CBA IPI.  
The evaluation offers a framework for improvement planning and helps the organisation in understanding its current state of process maturity.
- iv. *SEI CMM as the Basis:* CBA IPI uses the SEI CMM as the foundation for the assessment. The CMM provides a set of best practices and guidelines for the purpose of establishing process maturity and competence in software engineering.

To evaluate the maturity of each area, CBA IPI incorporates the organization's processes with the CMM's process areas, goals, and practises.

- v. *Internal Process Improvement:* The primary objective of CBA IPI is to direct organisational efforts to enhance internal processes. The assessment results support the development of action plans to improve the organization's processes by identifying specific areas for evolve, prioritising improvement initiatives, and more.
- vi. *Continuous Improvement:* CBA IPI supports the concept of continuous improvement by enabling organizations to periodically reassess their processes and track progress over time. It encourages the organization to learn from the assessment findings, implement changes, and measure the effectiveness of improvement efforts.

### **3. SPICE (ISO/IEC15504)**

Software Process Improvement and Capability Determination, or SPICE, is a global standard with the ISO/IEC 15504 code. It offers a set of requirements and guidelines for evaluating the productivity of software processes within firms. The major goals of SPICE are to assist organisations in conducting unbiased assessments of their software processes, identify problem areas, and increase overall process capabilities.

### **4. ISO 9001:2000 for Software:**

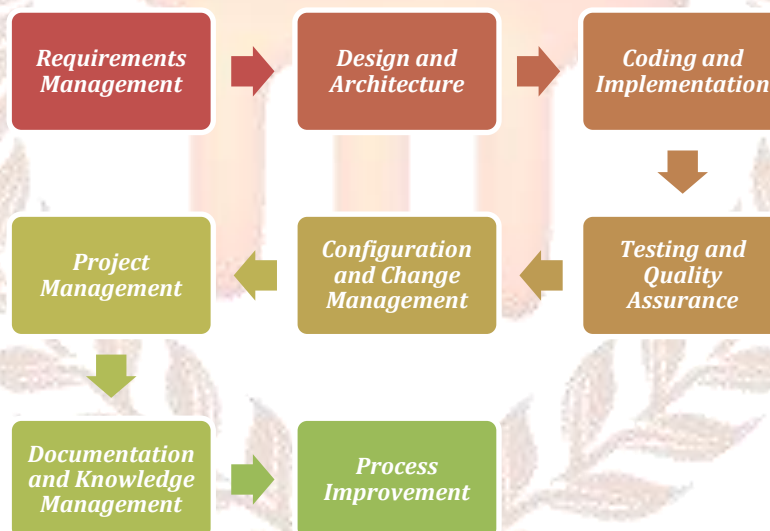
An international standard called ISO 9001:2000 is a general one that can be used by any organisation that wants to raise the overall quality of its systems, goods, or services. While it is not specifically designed for software organizations, it is directly applicable to them. The standard gives organisations a framework for applying a quality management system (QMS), guaranteeing that they satisfy customer needs and improve their processes and product.

### 3. DIMENSIONS OF PROCESS:

Several factors are considered while assessing the process life-cycle models used in software engineering to determine the efficiency and maturity of the software development processes.

These dimensions offer a thorough overview of the different aspects of the processes being evaluated.

Although the dimensions of different assessment models may vary, the following are some commonly accepted dimensions in the evaluation of process life-cycle models in software engineering:



- Requirements Management:** This dimension focuses on assessing how well the organisation can manage software requirements like requirement elicitation, requirement documentation, requirement validation, and requirement traceability.
- Design and Architecture:** This dimension evaluates the organization's methods for developing and implementing software considering elements like component design, interface design, and the application of design patterns and concepts, as well as architectural design.
- Coding and Implementation:** This dimension focuses on the organization's coding and implementation procedures. It includes factors like Coding standards, readability and

maintainability of the code, adherence to best coding practices, and the use of suitable programming languages and tools.

- d. **Testing and Quality Assurance:** This dimension focuses on assessing the organization's quality control and testing procedures. It includes tasks such as test case design, execution, defect tracking, and the usage of automated testing tools.
- e. **Configuration and Change Management:** This dimension evaluates the organization's procedures for handling changes and managing software configuration. It considers factors like software release management, configuration management, change management procedures, and version control.
- f. **Project Management:** This dimension evaluates the organization's project management practices. It includes aspects like project planning, resource management, risk management, project tracking, and estimation and scheduling.
- g. **Documentation and Knowledge Management:** This component focuses on the organization's procedures for managing knowledge and documenting software assets. It includes factors like documentation standards, accuracy and completeness, knowledge-sharing systems, and the use of documentation tools.
- h. **Process Improvement:** This dimension examines the organization's efforts in continuously improving its software engineering processes. It includes aspects such as process assessment, process measurement, process tailoring, process training, and the adoption of process improvement frameworks or models.

#### 4. THE NEED FOR A BUSINESS MODEL IN SOFTWARE ENGINEERING.

The need for a business model arises from the realisation that software development is not only a technical endeavour but also a business activity in the assessment of process life-cycle models in software engineering. A business model provides a framework for understanding and evaluating how software processes and the organization's overall business goals are aligned. It helps in evaluating the software's value proposition, revenue sources, cost structure, and other important factors.

Here are a few reasons why a business model is essential when evaluating process life-cycle models in software engineering:

- a. **Alignment with Business Goals:** A business model ensures that software development processes are aligned with the organization's overall business objectives. It helps evaluate how well the software development processes support the organization's strategic goals, market positioning, and customer needs.
- b. **Value Proposition:** Value Proposition refers to the unique value or benefits that a software development process offers to the organization and its stakeholders. It represents the reasons why the software development process is valuable and advantageous compared to alternative approaches or competitors.  
It helps in determining whether the software meets market demand, offers special features or advantages, and adds value for users and stakeholders.
- c. **Revenue Generation:** It refers to the ability of software development processes to generate income or financial value for the organization by considering the aspects like pricing strategies, licensing models, sales channels, and the monetization potential of the software product or service.
- d. **Cost Structure:** It refers to the breakdown and analysis of costs associated with the software development process considering the factors like refer to the breakdown and analysis of costs associated with the software development process.
- e. **Market Analysis:** It refers to the systematic evaluation and understanding of the market dynamics and factors that influence the success of software products or services. It helps evaluate how well the software development processes consider market trends, customer preferences, and competitive advantages.



- f. **Risk Assessment:** It refers to the systematic evaluation and analysis of potential risks or uncertainties associated with software development processes. It involves identifying, analysing, and prioritizing risks to determine their potential impact on the success of the processes and the overall project and it helps identify and evaluate risks associated with market acceptance, technology adoption, regulatory compliance, and other factors that may impact the success of the software product or service.
- g. **Business Continuity:** It refers to the ability of an organization to maintain the uninterrupted operation of its software development processes and services in the face of disruptive events or circumstances. It involves implementing strategies, plans, and measures to ensure the availability, resilience, and recovery of critical processes and resources in the event of disruptions, such as natural disasters, technology failures, or human errors.

## 5. SUMMARY

The assessment of the process in software engineering involves evaluating the effectiveness and maturity of software development processes to identify areas for improvement. It helps organizations understand their strengths and weaknesses and provides a basis for implementing process improvements.

Software process assessment can be conducted using various approaches like SCAMPI, CBA IPI, ISO/IEC 15504, and ISO 9001:2000 for Software.

The assessment of the process considers multiple dimensions to evaluate different aspects of software development. These dimensions include requirements management, design and architecture, coding, and implementation, testing and quality assurance, configuration and change management, project management, documentation and knowledge management, measurement and metrics, and process improvement.

A business model helps assess the alignment of software development processes with the organization's overall business objectives, value proposition, revenue generation, cost structure, market analysis, risk assessment, and business continuity. It ensures that software processes support the organization's strategic goals, market needs, and create value for customers and stakeholders.

## 6. SELF-ASSESSMENT QUESTIONS

1. The main purpose of assessing the software process?
  - a) To measure the performance of individual developers
  - b) To evaluate the effectiveness and maturity of software development processes
  - c) To ensure compliance with coding standards
  - d) To determine software licensing models
2. Which statement best describes the role of process assessments in software engineering?
  - a) To validate user requirements
  - b) To verify the correctness of software code
  - c) To improve the efficiency and effectiveness of software development processes
  - d) To determine software pricing and licensing models
3. Which of the following is a common dimension assessed in software process evaluations?
  - a) Technical skills and qualifications of the development team
  - b) Software licensing and copyright compliance
  - c) Requirements management, design and architecture, coding and implementation, etc.
  - d) Market share and competition analysis
4. Which approach uses a maturity model to assess the organization's software processes?
  - a) CMM-Based Appraisal for Internal Process Improvement (CBA IPI)
  - b) Standard CMMI Assessment Method for Process Improvement (SCAMPI)
  - c) ISO/IEC 15504 (SPICE)
  - d) Lean Six Sigma
5. Which dimension assesses the organization's ability to manage software requirements effectively?
  - a) Requirements Management
  - b) Design and Architecture
  - c) Testing and Quality Assurance

- d) Project Management
6. Why is a business model important in the assessment of software development processes?
    - a) To measure the code quality of the software
    - b) To ensure compliance with industry standards
    - c) To evaluate the alignment with business objectives and value creation
    - d) To identify potential bugs or errors in the software
  7. How does a business model contribute to software process assessment?
    - a) It helps in debugging and fixing software defects.
    - b) It ensures adherence to coding standards and best practices.
    - c) It aligns software development with overall business objectives and value creation.
    - d) It tracks project schedules and resource utilization.
  8. The main purpose of a business model in software engineering is
    - a) To manage software licenses and copyrights.
    - b) To analyse market trends and competition.
    - c) To ensure compliance with industry regulations.
    - d) To provide a framework for generating revenue and creating value.
  9. Which dimension evaluates the effectiveness of project planning and control?
    - a) Requirements Management
    - b) Design and Architecture
    - c) Project Management
    - d) Testing and Quality Assurance
  10. Which dimension focuses on evaluating the effectiveness of software testing techniques and defect management processes?
    - a) Requirements Management
    - b) Design and Architecture
    - c) Testing and Quality Assurance
    - d) Project Management

## 7. SELF-ASSESSMENT ANSWERS

1. B - To evaluate the effectiveness and maturity of software development processes
2. C- To improve the efficiency and effectiveness of software development processes.
3. C- Requirements management, design and architecture, coding, and implementation, etc.
4. A- CMM-Based Appraisal for Internal Process Improvement (CBA IPI)
5. A- Requirements Management
6. C- To evaluate the alignment with business objectives and value creation.
7. C - It aligns software development with overall business objectives and value creation.
8. D- To provide a framework for generating revenue and creating value.
9. C - Project Management
10. C - Testing and Quality Assurance.

## 8. TERMINAL QUESTIONS

1. Elucidate the software engineering assessment procedure in detail.
2. Briefly explain the different approaches to software process assessment and its improvement.
3. Briefly explain the important concepts to comprehend CMM-Based Appraisal for Internal Process Improvement (CBA IPI).
4. Explicate the different dimensions in the evaluation of process life-cycle models.
5. How does a business model contribute to aligning software development with overall business objectives?



## 9. TERMINAL ANSWERS

1. Refer to section 6.2.
2. Refer to section 6.2.1
3. Refer to section 6.2.1
4. Refer to section 6.3.
5. Refer to section 6.4.

