Software Process & Project Management

Chapter Two :

Software Process Assessment and Improvement

Outline

Software Process Assessment and Improvement

2.1 Software Process Assessment Model

2.2 Software Process Assessment Method

2.3 Software Process Improvement Model

2.4 Continuous and staged software process rating

Software Process Assessment and Improvement (SPAI)

- A systematic approach to evaluating and enhancing software development processes.
- Structured approach to implementing changes to software processes to make them more effective and efficient
- Aim to optimize processes for better quality, efficiency, and alignment with business goals.

- Purpose.
  - Identify areas where processes are effective.
  - Pinpoint areas where processes are inefficient or ineffective.
  - Provide a basis for process improvement initiatives.
  - Benchmark against industry best practices..

Software Process Assessment

A software process assessment is a disciplined examination of the software processes used by an organization, based on a process model. The assessment includes the identification and characterization of current practices, identifying areas of strengths and weaknesses, and the ability of current practices to control or avoid significant causes of poor (software) quality, cost, and schedule.

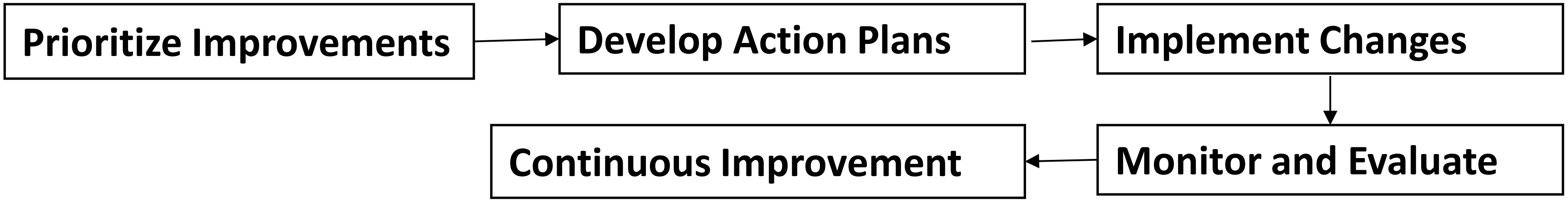
A software assessment (or audit) can be of three types :

- **A self-assessment (first-party assessment)** is performed internally by an organization's personnel.
- **A second-party assessment** is performed by an external assessment team or the organization is assessed by a customer.
- **A third-party assessment** is performed by an external party or (e.g., a supplier being assessed by a third party to verify its ability to enter contracts with a customer).

Software process assessments are performed in an open and collaborative environment. They are for the use of the organization to improve its software processes, and the results are confidential to the organization. The organization being assessed must have members on the assessment team.

The Improvement Process:

- Prioritize Improvements: prioritize areas for improvement.
- Develop Action Plans: Create detailed action plans for implementing the prioritized improvements.
- Implement Changes: Put the action plans into effect, making the necessary process changes.
- Monitor and Evaluate: Track the impact of the implemented changes and evaluate their effectiveness.
- Continuous Improvement: Iterate on the improvement process, making ongoing adjustments.



Improvement Techniques

Common Techniques:

Process Modelling: Visualizing and documenting processes.

Process Simulation: Simulating process changes to predict their impact.

Root Cause Analysis: Identifying the underlying causes of process problems.

Benchmarking: Comparing processes to industry best practices.

Training and Education: Providing training to ensure team members understand and can follow improved processes.

Cont’d

Benefits of Assessment and Improvement

- Improved Software Quality: Fewer defects and higher reliability.
- Increased Productivity: More efficient development processes.
- Reduced Costs: Lower development costs and rework.
- Enhanced Predictability: More accurate project estimates and schedules.
- Increased Customer Satisfaction: Delivering higher-quality products that meet customer needs..

2.1 Software Process Assessment Model

- A structured framework used to evaluate the maturity and capability of an organization's software development processes
- Provide a benchmark for organizations to understand their strengths and weaknesses,
- identify areas for improvement and guide process improvement initiatives.

Cont’d

Purpose of SPAMs:

- **Process Improvement:** To identify areas where software development processes can be improved to increase efficiency, quality, and predictability.
- **Benchmarking:** SPAMs allow organizations to compare their processes against industry best practices and other organizations.
  - SPAMs help mitigate software development risks by pinpointing process weaknesses.
  - SPAMs assess an organization's ability to produce quality software

Cont’d

Advantages of SPAM:

- Improved Software Quality
- Efficiency: Streamlined and optimized processes lead to increased efficiency in software development.
- Reduced Costs: By preventing defects and rework, organizations can reduce the costs associated with software development.
- Enhanced Customer Satisfaction: Higher quality products and predictable delivery schedules lead to increased customer satisfaction.

Limitations of SPAMs:

- Resource Intensive
- Complexity
- Resistance to Change.

2.2 Assessment Methods

Tools for Evaluation

- Document Review: Examining process documentation, plans, and reports.
- Interviews: Conducting structured interviews with key stakeholders.
- Surveys: Distributing questionnaires to gather feedback from a wider audience.
- Observations: Observing development teams in action.
- Metrics Analysis: Analyzing project data and metrics to assess process performance.

Here are some common methods used in Software Process Assessment:

1. Capability Maturity Model (CMM) : Description: A framework that helps organizations improve their software processes through **five maturity levels**.:
Initial, Managed, Defined, Quantitatively Managed, Optimizing

2. CMMI (Capability Maturity Model Integration) : An evolution of CMM, CMMI integrates various models and provides a comprehensive framework for process improvement across multiple disciplines.

3. SPICE (Software Process Improvement and Capability dEtermination)
An international standard (ISO/IEC 15504) for the assessment of software processes, focuses on process capability and performance.

4. ISO 9001: A standard that specifies requirements for a quality management system, emphasizing process improvement and customer satisfaction.

5. Agile Assessment: Tailored assessments for Agile methodologies, focusing on principles like flexibility, iterative development, and team collaboration.

Software Process Assessment Cycle

According to Paulk and colleagues (1995), the CMM-based assessment approach uses a six-step cycle. They are −

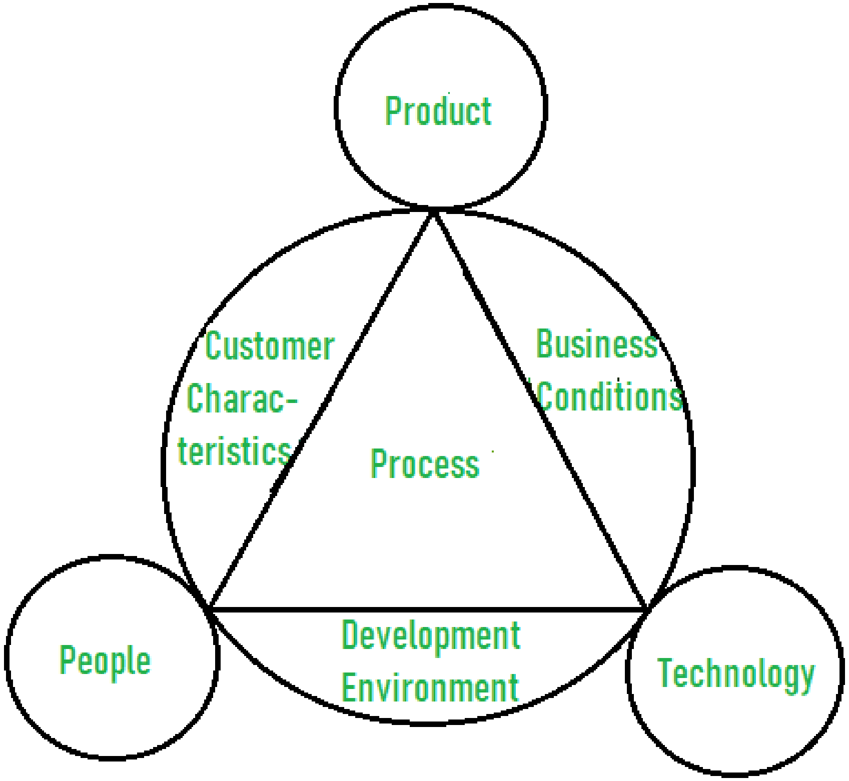
- Select a team - The members of the team should be professionals knowledgeable in software engineering and management.
- The representatives of the site to be appraised complete the standard process maturity questionnaire.
- The assessment team performs an analysis of the questionnaire responses and identifies the areas that warrant further exploration according to the CMM key process areas.
- The assessment team conducts a site visit to gain an understanding of the software process followed by the site.
- The assessment team produces a list of findings that identifies the strengths and weaknesses of the organization's software process.
- The assessment team prepares a Key Process Area (KPA) profile analysis and presents the results to the appropriate audience.

Software Process Assessment Cycle… Cont'd

- For example, the assessment team must be led by an authorized SEI Lead Assessor. The team must consist of between four to ten team members. At least, one team member must be from the organization being assessed, and all team members must complete the SEI's Introduction to the CMM course (or its equivalent) and the SEI's CBA IPI (**Sof**tware Engineering Institute's **CMM-Base**d Assessment for **Internal** **P**rocess Improvement) team training course. Team members must also meet some selection guidelines.
- With regard to data collection, the CBA IPI relies on four methods −
  1. The standard maturity questionnaire
  2. Individual and group interviews
  3. Document reviews
  4. Feedback from the review of the draft findings with the assessment participants

2.3 Software Process Improvement Framework

A "software process improvement framework" is a structured set of principles, methods, and tools designed to help organizations identify, analyze, and improve their software development processes, aiming to achieve higher quality, faster delivery times, and reduced costs by systematically optimizing their workflows and practices; popular examples include CMMI, Agile, and Six Sigma, each with its own specific approach to process improvement.



Key points about Software Process Improvement Frameworks:

Focus on continuous improvement: These frameworks encourage ongoing evaluation and refinement of software development processes, not just a one-time fix.

Defined stages or levels: Most frameworks consist of a series of maturity levels or stages that organizations can progress through, indicating increasing process capability.

Tailorable to specific needs: Frameworks are flexible enough to be adapted to the unique context and requirements of different organizations.

Metrics and measurements: Key performance indicators (KPIs) are used to track progress and identify areas for improvement within the software development process.

Common Software Process Improvement Frameworks :

Capability Maturity Model Integration (CMMI):
A widely recognized framework focusing on process improvement across various organizational disciplines, including software development.

Agile methodologies (Scrum, Kanban):
Emphasize iterative development, flexibility, and collaboration, with a focus on delivering value quickly and adapting to changing requirements.

Six Sigma:
A data-driven approach to process improvement, aiming to reduce defects and variability to near-zero levels.

Lean Software Development:
Based on lean manufacturing principles, aiming to eliminate waste and optimize the software development process.

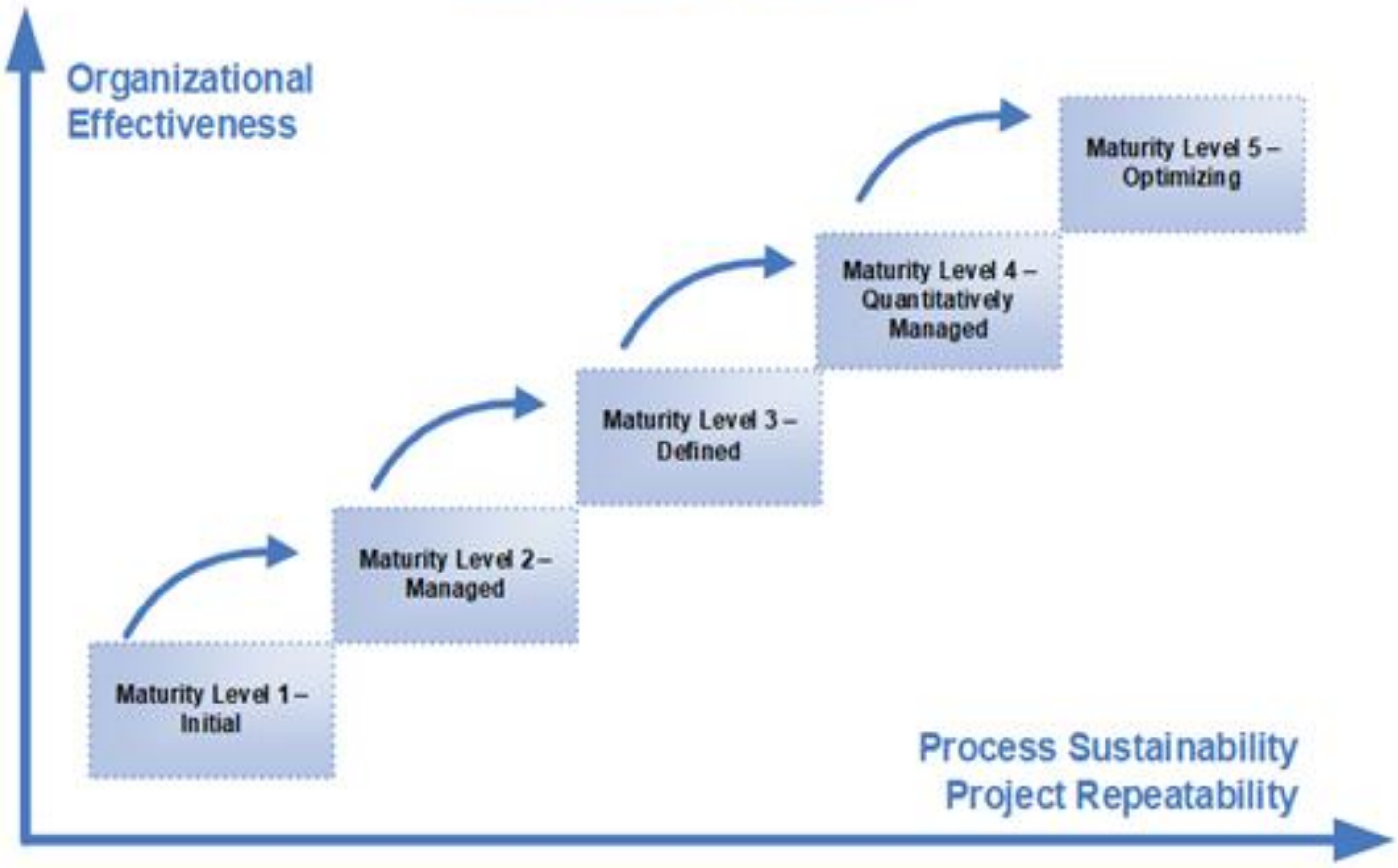
Benefits of using a software process improvement framework:

- **Increased quality:** By identifying and addressing process weaknesses, software quality can be significantly improved.
- **Improved project predictability:** More consistent processes lead to better project planning and estimations.
- **Faster time to market:** Streamlining development processes can accelerate delivery times.
- **Reduced costs:** Efficient processes can minimize rework and resource waste.

2.4 Continuous and Staged Software Process Rating

In software development, a "continuous process rating" means evaluating each individual process within an organization separately, assigning a maturity level to each one based on its own performance, while a "staged process rating" means assigning a single maturity level to all processes within a specific level, essentially grouping them together based on their overall maturity level, typically within a framework like the Capability Maturity Model Integration (CMMI).

CMMI Maturity Levels


Organizational Effectiveness
Maturity Level 1 – Initial
Maturity Level 2 – Managed
Maturity Level 3 – Defined
Maturity Level 4 – Quantitatively Managed
Maturity Level 5 – Optimizing
Process Sustainability
Project Repeatability

Key Differences:

Granularity:

Continuous rating provides a more detailed view by assessing each process individually, whereas staged rating provides a broader picture by evaluating groups of processes at a single maturity level.

Flexibility:

Continuous rating allows for more tailored improvement plans as each process can be addressed independently, while staged rating may require focusing on improving all processes within a specific level before moving to the next.

Example with CMMI:

Continuous:

A company might assess its "Requirement Management" process as being at Capability Level 3 (Defined), while its "Risk Management" process is at Level 2 (Managed).

Staged:

If the company is considered at Maturity Level 2 (Managed), all its processes would be evaluated against the requirements for Level 2, regardless of individual process performance.

When to Use Each Approach:

- **Continuous:**

When a company wants to identify specific areas for improvement within their processes and focus on targeted optimization.

- **Staged:**

When a company is just starting its process improvement journey and needs a structured path to follow, focus on achieving a general level of maturity across all processes before diving deeper.