Chapter 6: Software Process Improvement

Software Process Improvement

6.1 The Rationale for Process Improvement

6.2 Software Process Improvement (SPI) Cycle

6.3 Process Exceptions

6.4 The CMMI Process Improvement Framework

6.1 The Rationale for Process Improvement

- Why Process Improvement
- Quality Factors
- Software Process Improvement Considerations
- Approaches to Software Process Improvement

Why Process Improvement?

6.1 The Rationale for Process Improvement

Achieves benefits

Enhance Quality

Reduce development Cost

reduce Development time

- A good process is usually required to produce a good product, and process improvement benefits arise because the quality of the product depends on its development process.
- Hence, software companies turn to software process improvement with the hopes of achieving the following benefits:
  - ✧To reduce development time.
  - ✧To reduce development costs
  - ✧To increase the quality of software

Quality Factors

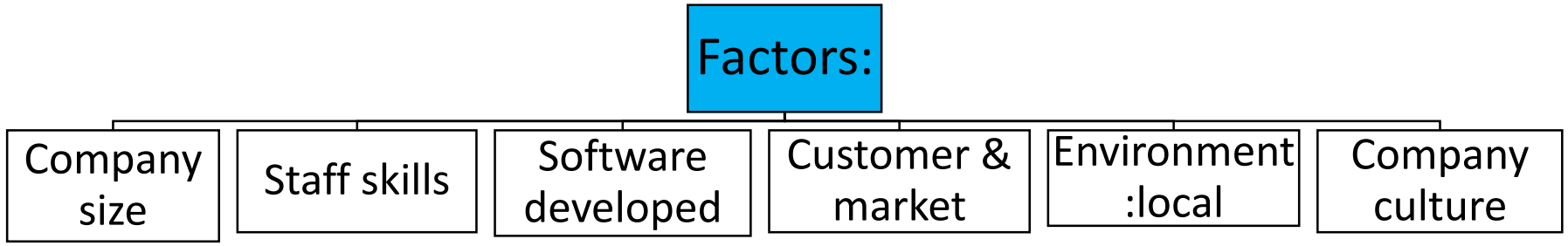
6.1 The Rationale for Process Improvement

- For manufactured goods, process is the principal quality determinant.
- For design-based activities, other factors are also involved, especially the capabilities of the designers.
- For large projects with ‘average’ capabilities, **the development process determines product quality.**
- For small projects, product quality depends on:
  - **the capabilities of the developers and**
  - **the development technology**

- **However, if an unrealistic schedule is imposed then all projects will suffer.**

Software Process Improvement Considerations

6.1 The Rationale for Process Improvement



- There is no one-size-fits-all or ‘ideal’ or ‘standard’ software process.
- Each company has to develop its own process depending on Factors:
  - company’s size : Small & Medium size (departments) , Retails outlet (Branches)
  - staff’s background and skills :Experienced, knowledge, skills
  - type of software being developed : Online based, Mobile App
  - customer and market requirements : Needs, Demand,
  - local environment
  - company culture: Quality culture

Software Process Improvement Considerations

6.1 The Rationale for Process Improvement

- For process improvement, you also have to consider what aspects of the process you want to improve, e.g.:
  - If the goal is to improve software quality, you may wish to introduce new process activities that change the way software is developed and tested;
  - If you are interested in improving some attribute of the process itself (e.g. development time), you have to decide which process attributes are the most important to the company.

Software Process Improvement Considerations

6.1 The Rationale for Process Improvement

Goal : improve current project's progress visibility

Process attributes: Visibility, Measurability

Process Characteristic / Attribute

- Understandability
- Measurability
- Reliability
- Robustness
- Standardization
- Supportability
- Maintainability
- Visibility
- Acceptability
- Rapidity

Software Process Improvement Considerations

6.1 The Rationale for Process Improvement

Process Characteristic / Attribute

Process characteristic/ attribute	Key issues
Understandability	To what extent is the process explicitly defined and how easy is it to understand the process definition?
Standardization	To what extent is the process based on a standard generic process? This may be important for some customers who require conformance with a set of defined process standards. To what extent is the same process used in all parts of a company?
Visibility	Do the process activities culminate in clear results, so that the progress of the process is externally visible?
Measurability	Does the process include data collection or other activities that allow process or product characteristics to be measured?
Supportability	To what extent can software tools be used to support the process activities?

Software Process Improvement Considerations

6.1 The Rationale for Process Improvement

Process Characteristic / Attribute

Process characteristic/ attribute	Key issues
Acceptability	Is the defined process acceptable to and usable by the engineers responsible for producing the software product?
Reliability	Is the process designed in such a way that process errors are avoided or trapped before they result in product errors?
Robustness	Can the process continue in spite of unexpected problems?
Maintainability	Can the process evolve to reflect changing organizational requirements or identified process improvements?
Rapidity	How fast can the process of delivering a system from a given specification be completed?

Approaches to Software Process Improvement

6.1 The Rationale for Process Improvement

Process Maturity

Agile approach

- The *process maturity* approach
  - Focuses on improving process and project management and introducing good software engineering practice.
  - The level of process maturity reflects the extent to which good technical and management practice has been adopted in organizational software development processes
  - e.g., the CMMI Process Improvement Framework
- The *agile* approach
  - Focuses on iterative development and the reduction of overheads in the software process
  - The primary characteristics of agile methods are rapid delivery of functionality and responsiveness to changing customer requirements
  - e.g., the Retrospective meetings in each sprint in the Scrum framework

6.2 Software Process Improvement Cycle

- Process Measurement
- Process Analysis
- Process Change

6.2 Software Process Improvement Cycle

Process measurement

Measure the attributes of current process. These measurement will be the baseline for assessing improvements.

Quantitative process data should be collected
Process metrics : Time, Resource (chap4)
Goal Question Metric (GQM) proposal

Note:
After Process Change, Re-measure the same attribute!

understand the relationships between the process activities and the measurements

Process analysis

Analyse current process to identify bottlenecks and weaknesses.

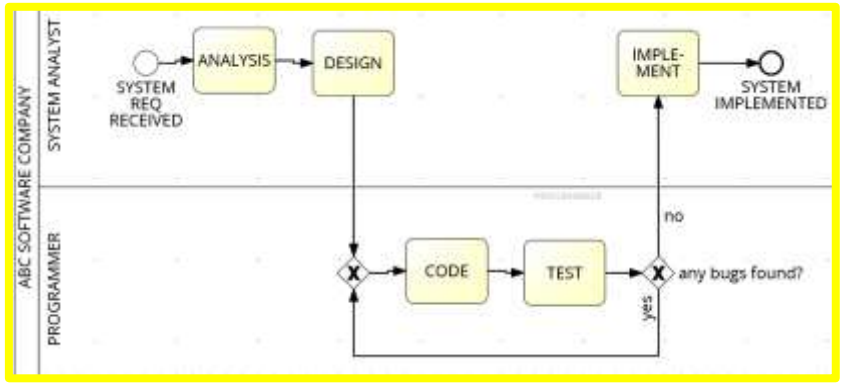
(note: **Process models** may be develop during this stage

Process change

Change the process to reduce/eliminate bottlenecks and weaknesses

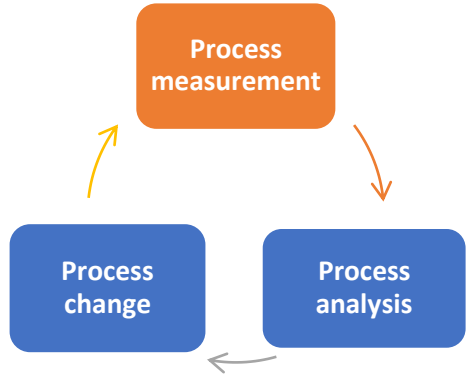
5 stages
Resistance Change,
Change persistence

Process model



Process Measurement

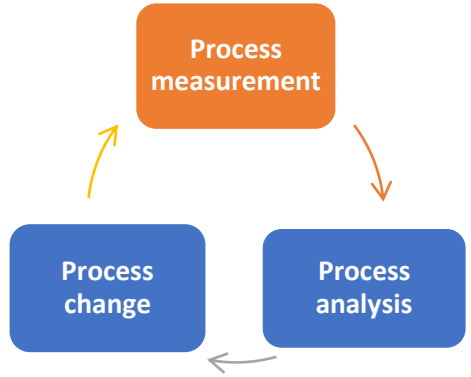
6.2 Software Process Improvement Cycle



- Wherever possible, quantitative process data should be collected
  - A clearly defined process is required in order to determine what to measure.
- Process measurements should be used to assess process improvements
  - But this does not mean that measurements should drive the improvements.
  - The improvement driver should be the organizational objectives.

Process Measurement

6.2 Software Process Improvement Cycle



Process Metrics Collection

3 types of process metrics can be collected during process measurement:

✧Time taken for process activities to be completed

- e.g. Calendar time to complete an activity or process.

✧Resources required for processes or activities

- e.g. Total effort in person-days, travel cost, computer resources.

✧Number of occurrences of a particular event

- e.g. Number of defects discovered.

Directly related to measure whether a process has been improved.
e.g. how much time and effort saved when moving from one fixed-point to another.

Directly related to software quality

Process Measurement

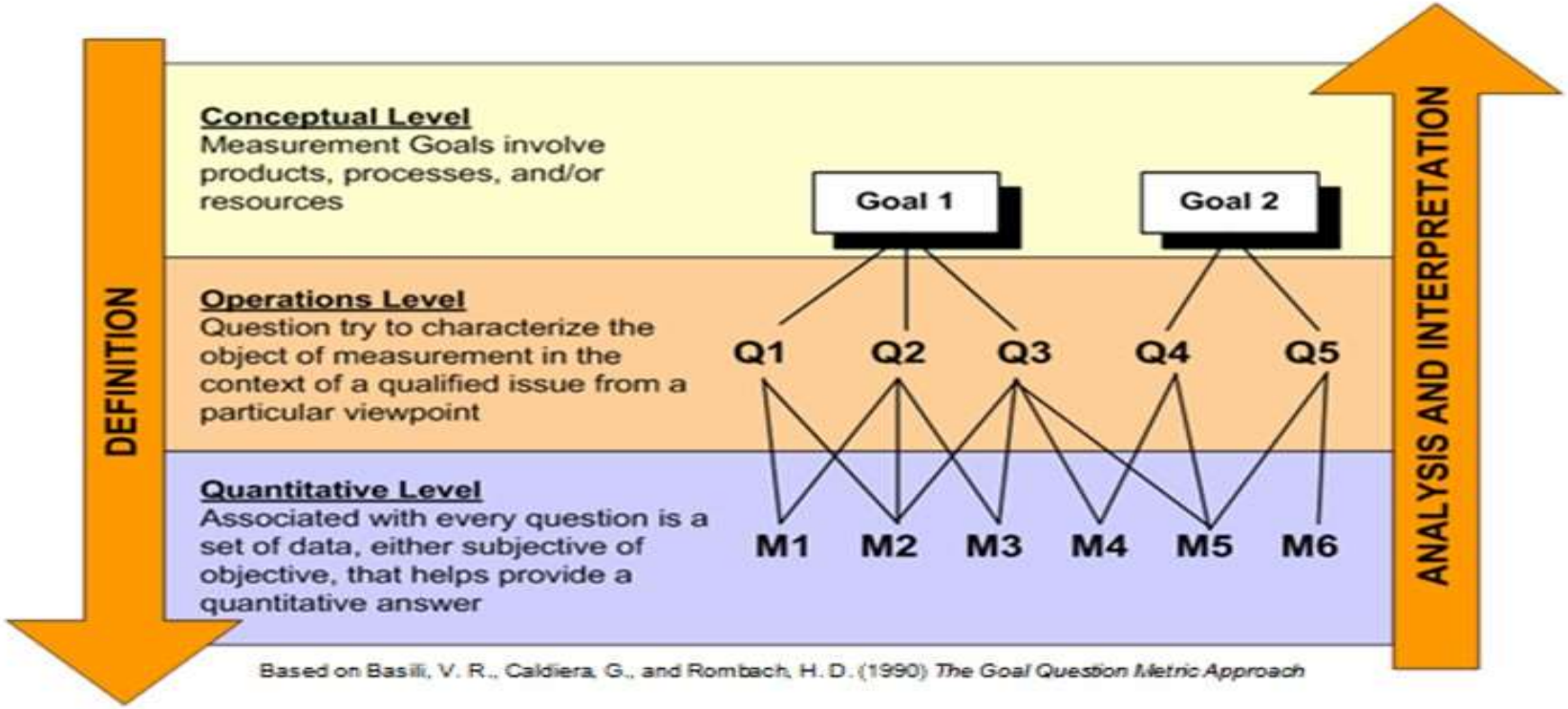
6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm

Difficulty in process measurement:

What information about the process should be collected to support process improvement?

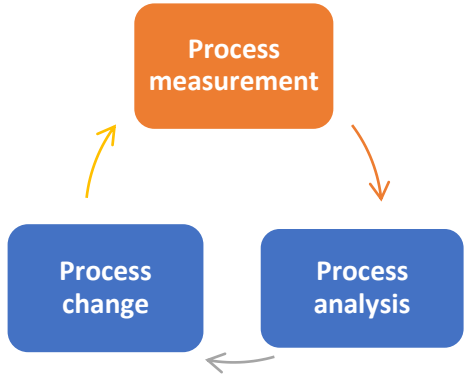
GQM is used to solve this difficulty.



Based on Basili, V. R., Caldiera, G., and Rombach, H. D. (1990) *The Goal Question Metric Approach*

Process Measurement

6.2 Software Process Improvement Cycle



Goal-Question-Metric (GQM) Paradigm Usage

The GQM paradigm is used in process improvement to answer 3 critical questions:

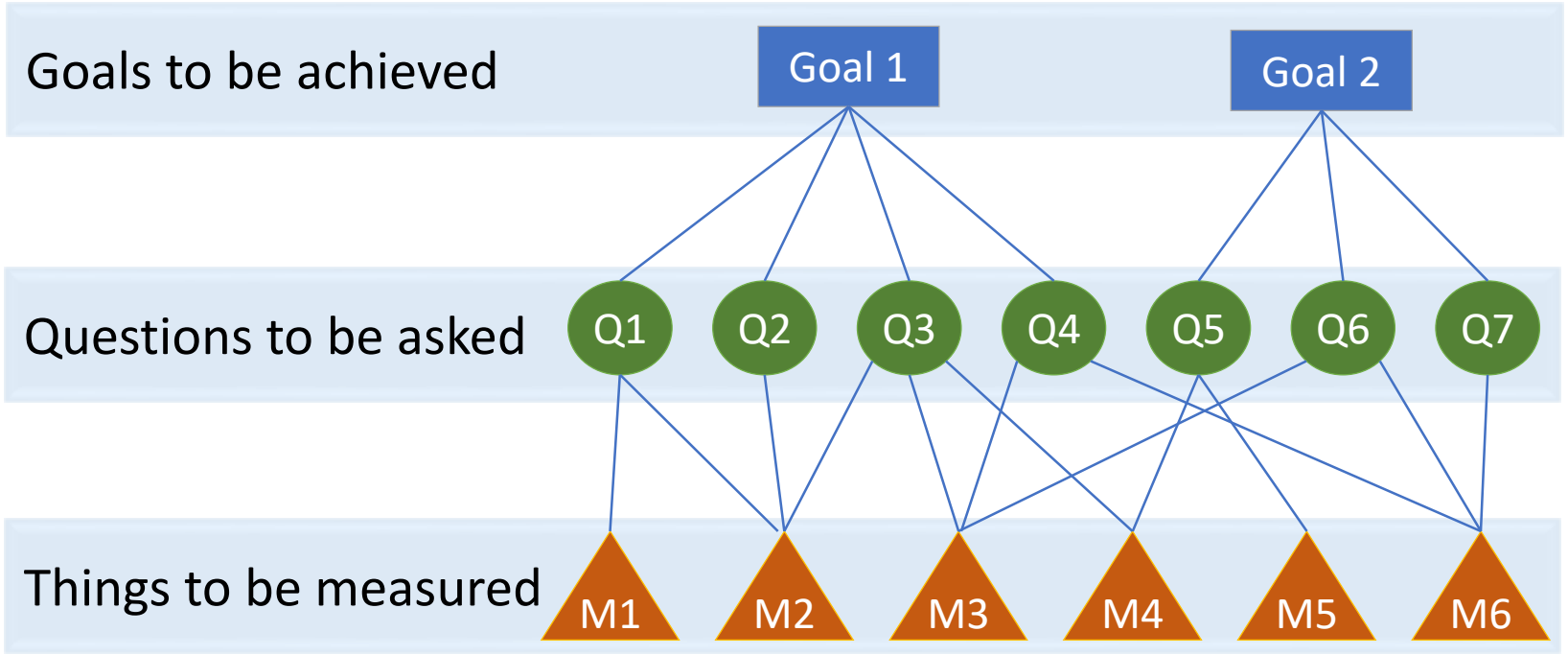
Goal	• Why are we introducing process improvement?
Questions	• What information do we need to help identify and assess improvements?
Metrics	• What process and product measurements are required to provide this information?

Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm Usage

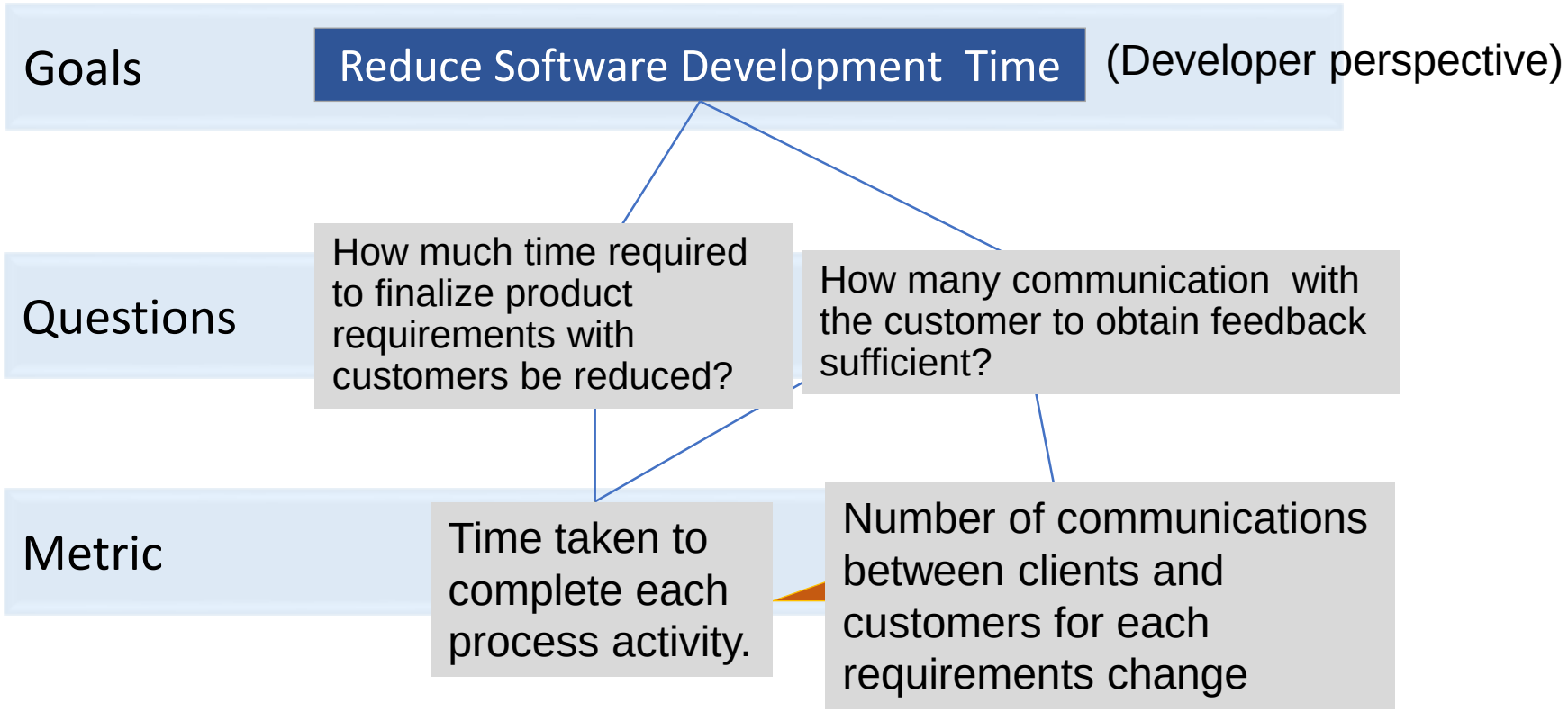
Use GQM to determine what information about the process should be collected to support process improvement



Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm – Example 1



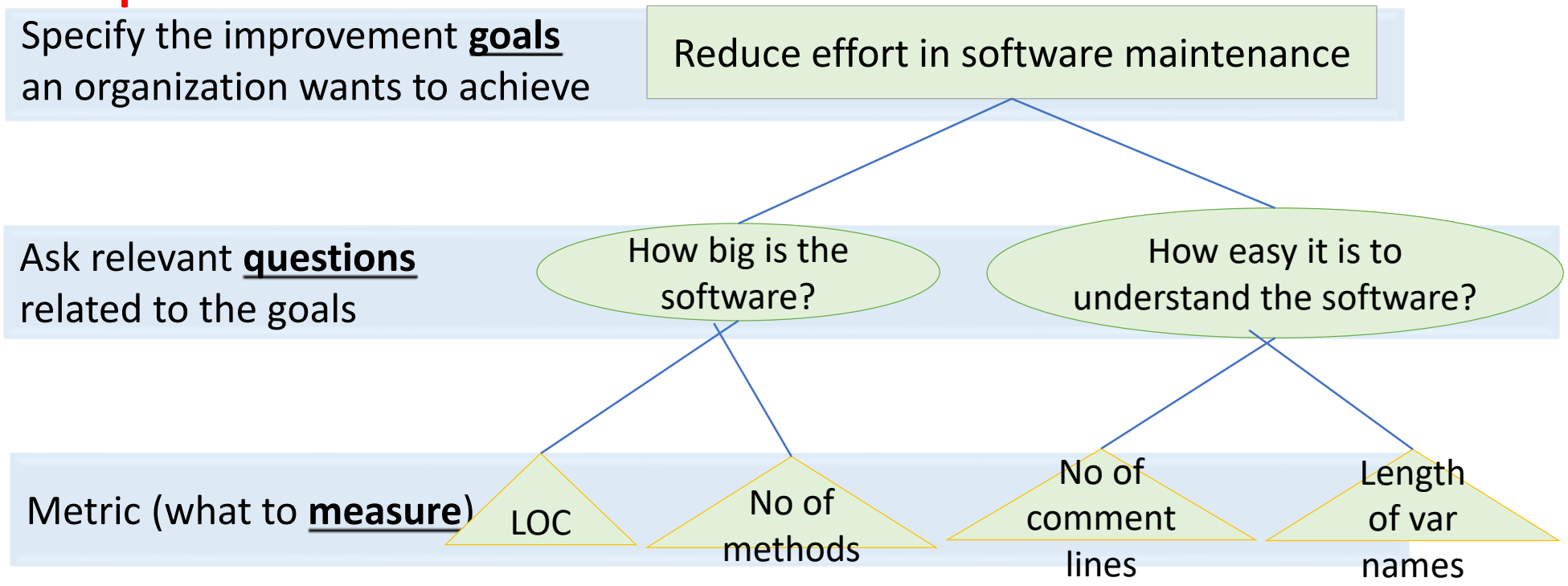
Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm

Use GQM to determine <u>what</u> information about the process should be collected to support process improvement

Example 2



Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm

Use GQM to determine what information about the process should be collected to support process improvement

Example 3

Specify the improvement **goals** an org^{n} wants to achieve

Improve the performance of an e-commerce website

Ask relevant **questions** related to the goals

What is the current page load time for the website?

How many users abandoning their carts before completing a purchase?

Metric (what to **measure**)

Page load time (e.g. T1)

cart abandonment rate (e.g. 20/day)

Q) Suggest how to reduce page load time? Use smaller pic size

Q) Suggest how to reduce cart abandonment? Consider carrying out usability test before system roll-out

GQM can help you to determine what information should be collected for process improvement

Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm – Component

- **Goals**
  - What is the organisation trying to achieve? The objective of process improvement is to satisfy these goals.
  - Focus on how the process affects products or the organization itself.
  - Example
    - improved level of process maturity,
    - shorter product development time,
    - increased product reliability.

Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm – Component

- **Questions**
  - Questions about areas of uncertainty related to the goals.
  - Example of Questions related to the goal of *shortening product development times*:
    - Where are the bottlenecks in our current process?
    - How can the time required to finalize product requirements with customers be reduced?

Process Measurement

6.2 Software Process Improvement Cycle

Goal-Question-Metric (GQM) Paradigm – Component

- **Metrics**
  - Collected to answer the questions and to confirm whether or not process improvements have achieved the desired goal.
  - **Examples** of measurements to be taken
    - time taken to complete each process activity (normalized by system size)
    - number of formal communications between clients and customers for each requirements change
    - number of defects discovered per test run.

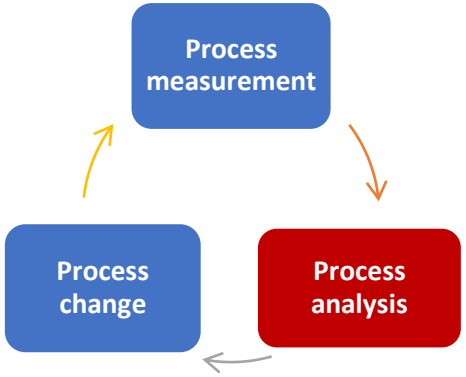
Goal-Question-Metric (GQM) Paradigm – Advantages

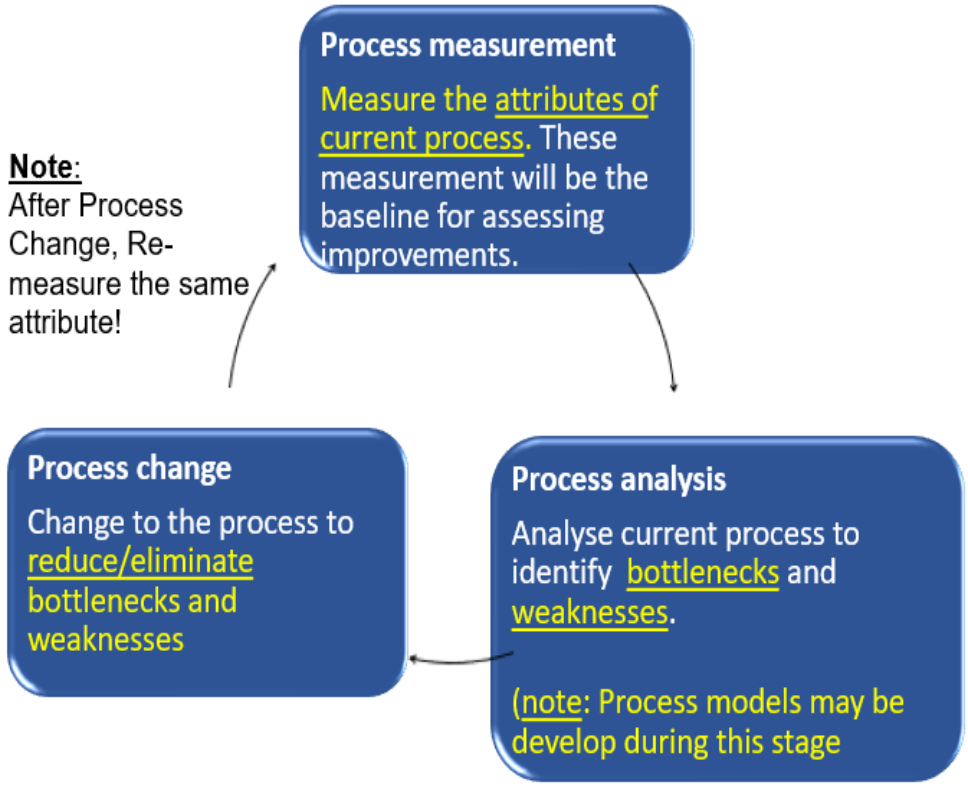
- Separates organizational concerns (goals) from specific process concerns (questions)
- Provides a basis for deciding what data should be collected.
- Suggests collect data should be analyzed in different ways, depending on the question it is intended to answer.

Process Analysis

6.2 Software Process Improvement Cycle

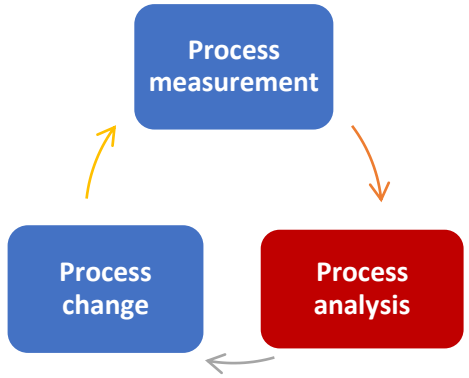
- The study of existing processes to understand the relationships between parts of the process and to compare them with other processes.
- Process analysis and process measurement are intertwined.
  - ➢ You need to carry out some analysis to know what to measure, and, when making measurements, you inevitably develop a deeper understanding of the process being measured.





Process Analysis

6.2 Software Process Improvement Cycle



Objectives

- To understand the activities involved in the process and the relationships between these activities.
- To understand the relationships between the process activities and the measurements that have been made.
- To relate the specific process or processes that you are analyzing to comparable processes elsewhere in the organization, or to idealized processes of the same type.

Techniques

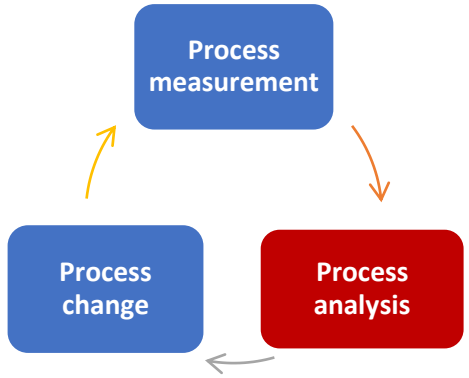
a. **Questionnaires and interviews**

b. **Ethnographic analysis**

c. Published process models

Process Analysis

6.2 Software Process Improvement Cycle



Techniques

a. Questionnaires and interviews

b. Ethnographic analysis (observation)

Eg: Social Media Analytics

a. Published process models

Process Analysis - Techniques

6.2 Software Process Improvement Cycle

a. Questionnaires and interviews

- **Questionnaires**:
  - ✓ Advantage: Fast
  - ✘ Disadvantage:
    - if questions are inappropriate → incomplete/inaccurate understanding of the process
    - Participants (engineers and PM) treated it as assessment and give answer that they think you want to hear.
- **Interviews**:
  - ✓ Advantage: open-ended → more information
- Answers to a formal questionnaires are refined during personal interviews.

Process Analysis - Techniques

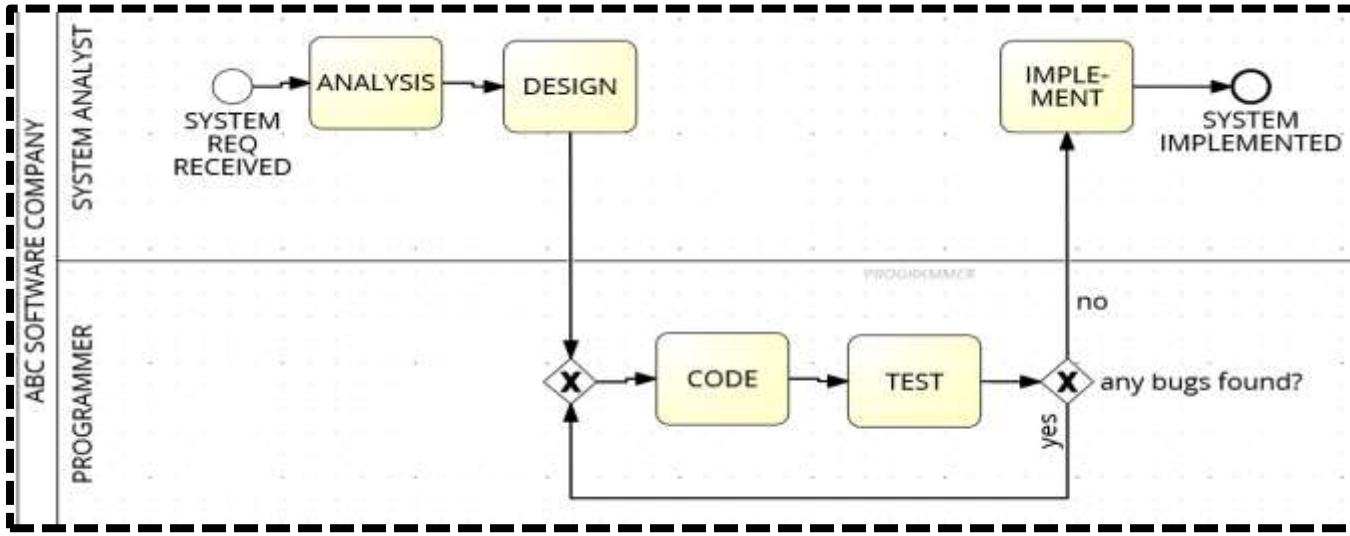
6.2 Software Process Improvement Cycle

b. Ethnographic analysis

- Involves assimilating process knowledge by observation.
- Best for in-depth analysis of process fragments rather than for whole-process understanding.
- The process fragments are chosen from the interview
- Disadvantage:
  - ✘ pro-longed process (several months) → from initial stages to maintenance.
  - ✘ Impractical for large projects (several years!!??)

Process Analysis - Techniques

6.2 Software Process Improvement Cycle



c. Published process models

- Process models are a good way of focusing attention on the activities in a process and the information transfer between these activities.
- Process models do not have to be formal or complete – purpose is to provoke discussion rather than document the process in detail.
- Model-oriented questions can be used to help understand the process e.g.
  - What activities take place in practice but are not shown in the model?
  - Are there process activities, shown in the model, that you (the process actor) think are inefficient?
  - If something goes wrong, do you follow the model or take up emergency actions? Why or why not?
  - How's the communication? Who involved? What's the bottlenecks?
  - What's the tool support the activities in the model? Effective? How to improve the tool?

Process Analysis - Aspects

6.2 Software Process Improvement Cycle

Process aspect	Questions
Adoption and standardization	Is the process documented and standardized across the organization? If not, does this mean that any measurements made are specific only to a single process instance? If processes are not standardized, then changes to one process may not be transferable to comparable processes elsewhere in the company.
Software engineering practice	Are there known, good software engineering practices that are not included in the process? Why are they not included? Does the lack of these practices affect product characteristics, such as the number of defects in a delivered software system?
Organizational constraints	What are the organizational constraints that affect the process design and the ways that the process is performed? For example, if the process involves dealing with classified material, there may be activities in the process to check that classified information is not included in any material due to be released to external organizations. Organizational constraints may mean that possible process changes cannot be made.

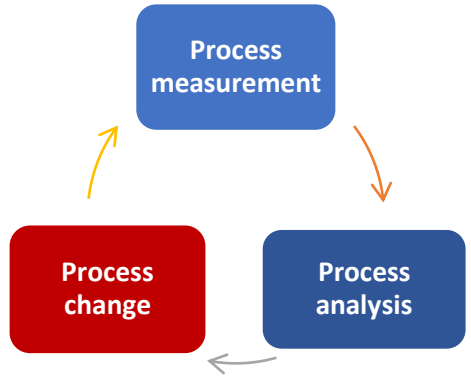
Process Analysis - Aspects

6.2 Software Process Improvement Cycle

Process aspect	Questions
Communications	How are communications managed in the process? How do communication issues relate to the process measurements that have been made? Communication problems are a major issue in many processes and communication bottlenecks are often the reasons for project delays.
Introspection	Is the process reflective (i.e., do the actors involved in the process explicitly think about and discuss the process and how it might be improved)? Are there mechanisms through which process actors can propose process improvements?
Learning	How do people joining a development team learn about the software processes used? Does the company have process manuals and process training programs?
Tool support	What aspects of the process are and aren't supported by software tools? For unsupported areas, are there tools that could be deployed cost-effectively to provide support? For supported areas, are the tools effective and efficient? Are better tools available?

Process Change

6.2 Software Process Improvement Cycle



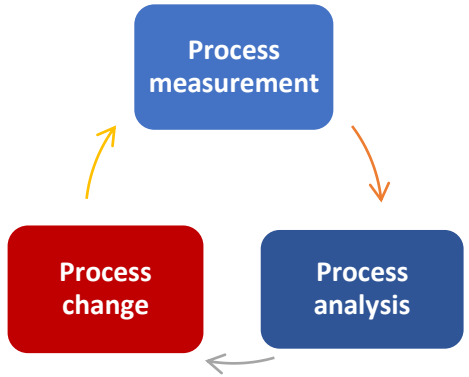
- Involves making modifications to existing processes.

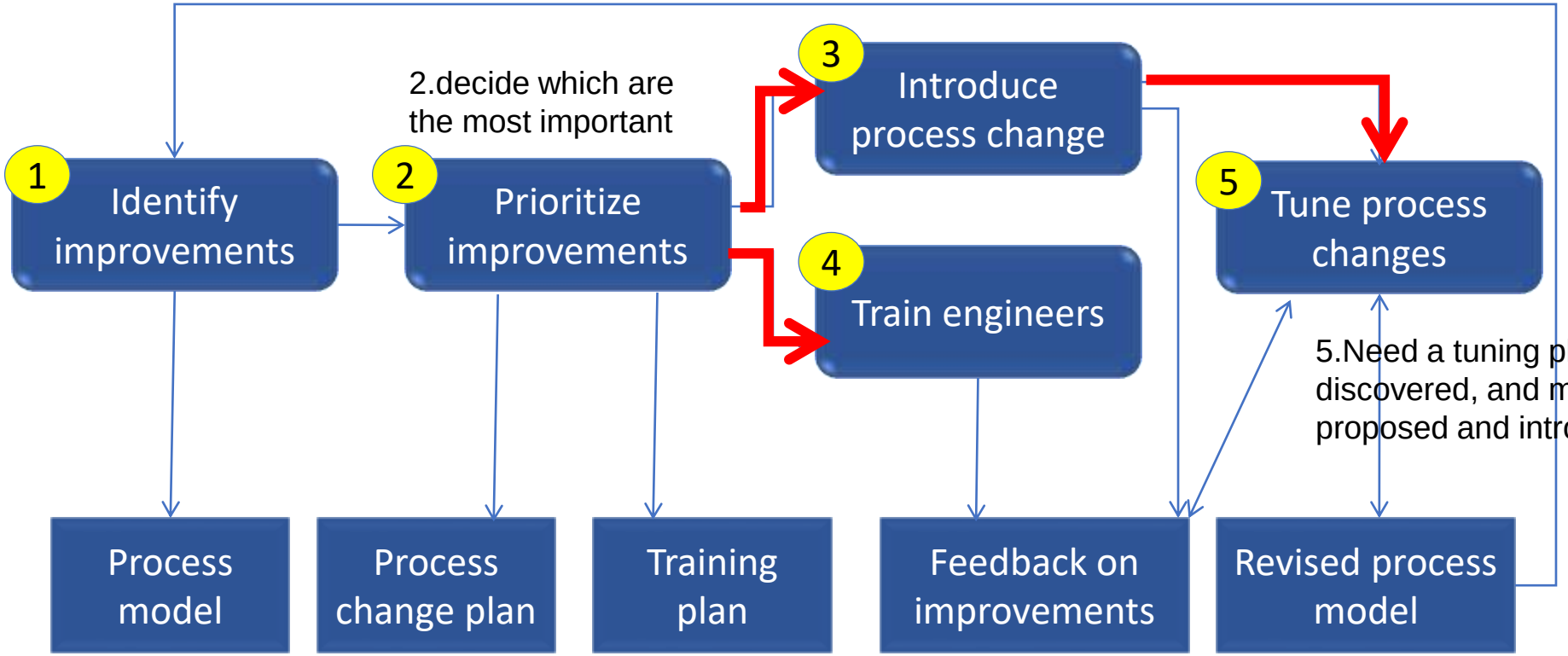
Factors caused the process change

- This may involve
  - Introducing new practices, methods or processes;
  - Changing the ordering of process activities;
  - Introducing or removing deliverables;
  - Introducing new roles or responsibilities.
- Change should be driven by measurable goals.

Process Change - Process

6.2 Software Process Improvement Cycle





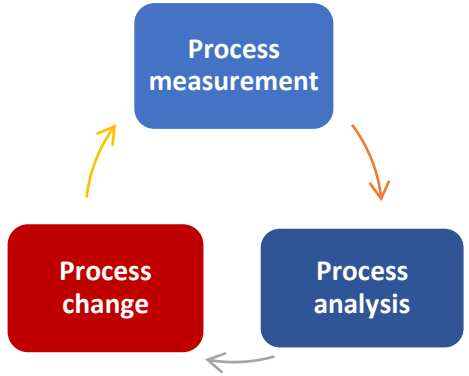
5.Need a tuning phase where minor problems can be discovered, and modifications to the process can be proposed and introduced.

1.use the results of the process analysis to identify ways to tackle quality problems, schedule / cost inefficiencies that have been identified during process analysis.

4. understand the changes that have been proposed and how to perform the new and changed processes

Process Change – 5 Stages

6.2 Software Process Improvement Cycle



1. **Improvement identification**: use the results of the process analysis to identify ways to tackle quality problems, schedule bottlenecks or cost inefficiencies that have been identified during process analysis.
2. **Improvement prioritization**: When many possible changes have been identified, it is usually impossible to introduce them all at once, and you must decide which are the most important.
3. **Process change introduction**: put new procedures, methods and tools into place and integrating them with other process activities.
4. **Process change training**: needed to gain the full benefits of process changes. The engineers involved need to understand the changes that have been proposed and how to perform the new and changed processes.
5. **Change tuning**: Proposed process changes will never be completely effective as soon as they are introduced. You need a tuning phase where minor problems can be discovered, and modifications to the process can be proposed and introduced.

Process Change – Difficulties

6.2 Software Process Improvement Cycle

1. Resistance to change

- Project managers may resist the introduction of process changes and propose reasons why changes will not work, or delay the introduction of changes. They may, in some cases, deliberately obstruct process changes and interpret data to show the ineffectiveness of proposed process change.

2. Change persistence

- While it may be possible to introduce process changes initially, it is common for process innovations to be discarded after a short time and for the processes to revert to their previous state.
- Introduce Tracking Attendance (Late)

Process Change – Difficulties

6.2 Software Process Improvement Cycle

1. Resistance to change

- **Project managers resist process change because any innovation has** unknown risks associated with it.
  - Project managers are judged according to whether or not their project produces software on time and to budget.
  - **PM may prefer an inefficient but predictable process** than an improved process that has organizational benefits, but which has short-term risks associated with it.
- **Engineers may resist the introduction of new processes for similar reasons or** because they see these processes as threatening their professionalism.
  - That is, **they may feel that the new process gives them less discretion and does not recognize the value of their skills and experience.**

Process Change – Difficulties

6.2 Software Process Improvement Cycle

2. Change Persistence

- The problem of changes being introduced then subsequently discarded
  - Changes may be proposed by an ‘evangelist’ who believes strongly that the changes will lead to improvement. He or she may work hard to ensure the changes are effective and the new process is accepted.
  - If the ‘evangelist’ leaves, then the people involved may therefore simply revert to the previous ways of doing things.
- Change institutionalization is important
  - process change is not dependent on individuals.
  - Organization should Institutionalize the *Change* i.e. make the change permanent by making it a standard practice in the company, with company-wide support and training.

6.3 Process Exceptions

- Software processes are complex and process models cannot effectively represent how to handle exceptions:
  - Several key people becoming ill just before a critical review;
  - A breach of security that means all external communications are out of action for several days;
  - Organisational reorganisation;
  - A need to respond to an unanticipated request for new proposals.

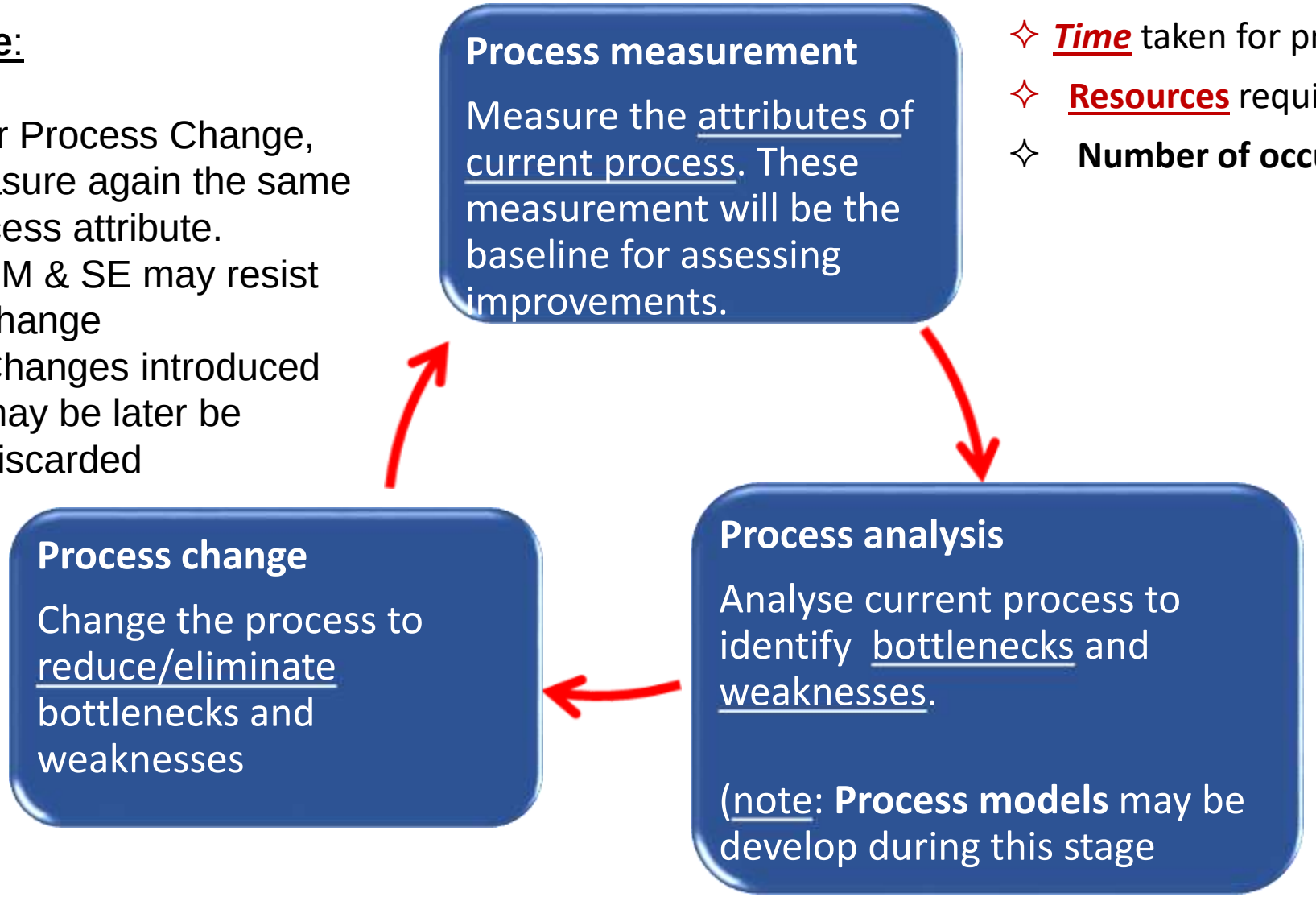
- Under these circumstances, the model is suspended and managers use their initiative to deal with the exception.
- There are too many exceptions cases, so Project Manager needs to be flexible to handle exception and dynamically change the “standard” process to cope with it.

6.2 Software Process Improvement Cycle – Brief Summary

Note:

After Process Change, measure again the same process attribute.

- PM & SE may resist change
- Changes introduced may be later be discarded



- ***Time*** taken for process activities to be completed
- **Resources** required for processes or activities
- **Number of occurrences of a particular event, etc**

6.4 The CMMI Process Improvement Framework

- CMMI Overview
- CMMI Process Areas
- CMMI Staged Model
- CMMI Continuous Model
- CMMI Continuous vs Staged Models

CMMI Overview

6.4 The CMMI Process Improvement Framework

- The Capability Maturity Model Integration (CMMI) Process Improvement Framework enable integration of source models such as the CMM for Software (SW-CMM), Integrated Product Development CMM (IPD-CMM), etc. developed by Carnegie-Mellon University's Software Engineering Institute (SEI)
- An evolutionary improvement path for software organizations and includes descriptions the key elements of an effective software process.
- Provides guidance on how to gain control of processes for developing and maintaining software.
- CCMI representations include:
  - **Staged Model/Representation (Maturity Levels)**
  - **Continuous Model/Representation (Capability Levels)**

CMMI Overview - Representation

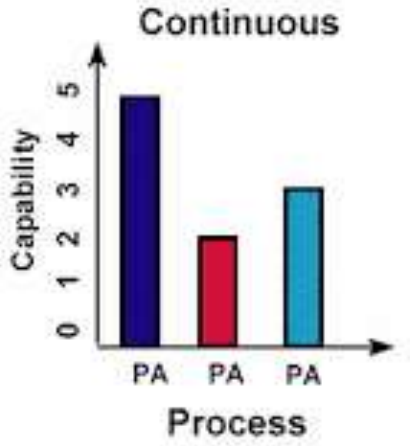
6.4 The CMMI Process Improvement Framework

a. CMMI Staged Model (Maturity Levels)

- Used in **Software** Capability Maturity Model (CMM)
- Uses predefined sets of process areas to define an improvement path for an organization.
- This improvement path is described by the Maturity Level model component, a well defined evolutionary towards achieving organizational processes.

b. CMMI Continuous Model (Capability Levels)

- Used in Systems Engineering Capability Model (SECM) and the IPD-CMM
- Allows an organization to select a specific process area and make improvements based on it.
- Uses Capability Levels to characterize improvement relative to an individual process area.



CMMI Process Areas

6.4 The CMMI Process Improvement Framework

- 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.
- The continuous representation enables the organization to choose the focus of its process improvement efforts by choosing those process that best benefit the organization and its business objectives.

- Each process area is defined by a set of **goals** and **practices**. There are two categories of goals and practices −
  - Generic goals and practices − They are a part of every process area.
  - Specific goals and practices − They are specific to a given process area.

(Refer to Appendix 6.1 : 10 pages)

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

CMMI Process Areas – Generic Goals & Practices

6.4 The CMMI Process Improvement Framework

- Generic goals (GG) & generic practices (GP) are a part of every process area.

GG 1 Achieve Specific Goals

GP 1.1 Perform Specific Practices

GG 2 Institutionalize a Managed Process

GP 2.1 Establish an Organizational Policy
GP 2.2 Plan the Process
GP 2.3 Provide Resources
GP 2.4 Assign Responsibility
GP 2.5 Train People
GP 2.6 Manage Configurations
GP 2.7 Identify and Involve Relevant Stakeholders
GP 2.8 Monitor and Control the Process
GP 2.9 Objectively Evaluate Adherence
GP 2.10 Review Status with Higher Level Management

GG 3 Institutionalize a Defined Process

GP 3.1 Establish a Defined Process
GP 3.2 Collect Improvement Information

GG 4 Institutionalize a Quantitatively Managed Process

GP 4.1 Establish Quantitative Objectives for the Process
GP 4.2 Stabilize Sub process Performance

GG 5 Institutionalize an Optimizing Process

GP 5.1 Ensure Continuous Process Improvement
GP 5.2 Correct Root Causes of Problems

CMMI Process Areas – Common Areas

6.4 The CMMI Process Improvement Framework

- The common features (attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting) are:

Commitment to Perform: Commitment to Perform describes the actions, the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship.

Ability to Perform: Ability to Perform describes the preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training.

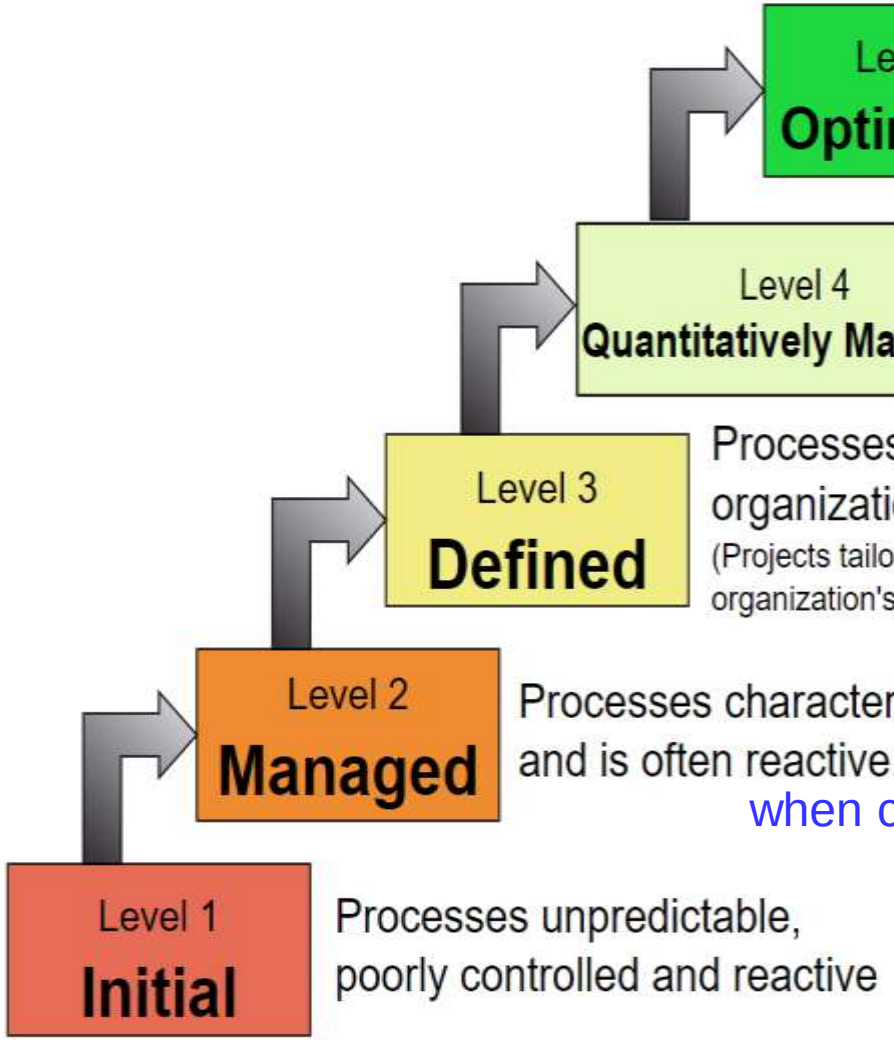
Activities Performed: Activities Performed describes the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.

Measurement and Analysis: Measurement and Analysis describes the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.

Verifying Implementation: Verifying Implementation describes the steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.

CMMI Staged Model - Maturity Levels Characteristics

6.4 The CMMI Process Improvement Framework



Focus on process improvement

Focuses on **continually improving process performance**

Processes measured and controlled

Quantitative objectives for quality and process performance are established and used as criteria in managing processes.

Processes characterized for the organization and is proactive.
(Projects tailor their processes from organization's standards)

Processes are **well-characterized and understood**, and are described in **standards**, **procedures**, **tools** and **methods**.

Processes characterized for projects and is often reactive.
when changes arise

organization has ensured that requirements are managed and that **processes are planned, performed, managed and controlled**.

Processes unpredictable, poorly controlled and reactive

CMMI Staged Model - Maturity Levels 1 Initial

6.4 The CMMI Process Improvement Framework

- Processes are usually **ad hoc** and **chaotic**.
- Success of the organization **depends on the competence of the people** in the organization and not on the use of proven processes.
- The organization usually **do not provide a stable environment**. Instead, software project success depends on having quality people.
- The organization often produce products and services that work. However, they **frequently exceed the budget and schedule** of their projects.
- Therefore, the organization is **characterized by:**
  - ✓ Tendency to over-commit
  - ✓ Abandon processes in the time of crisis
  - ✓ Not able to repeat their past successes

CMMI Staged Model - Maturity Levels 2 Managed

6.4 The CMMI Process Improvement Framework

- The organization has ensured that requirements are managed and that **processes are planned, performed, managed and controlled**.
- The process discipline reflected in this level helps to ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans.
- The status of the work products and the delivery of services are visible to management at defined points.
- Work products are reviewed with stakeholders and are controlled.
- The work products and services satisfy their specified requirements, standards and objectives.

CMMI Staged Model - Maturity Levels 3 Defined

6.4 The CMMI Process Improvement Framework

- Processes are **well-characterized and understood**, and are described in **standards**, **procedures**, **tools** and **methods**.
- Difference between level 2 and 3:
  - At level 2 the standards, process descriptions, and procedures may be quite different in each specific project. At level 3. the standards, process descriptions, and procedures for a project are tailored from the organization's set of standard processes to suit a particular project.
  - At level 3, processes are typically **described in more detail and more rigorously** than at level 2.
- At level 3, processes are managed more proactively using an understanding of the interrelationships of the process activities and detailed measures of the process, its work products and its services.

CMMI Staged Model - Maturity Levels 4 Quantitatively Managed

6.4 The CMMI Process Improvement Framework

- **Subprocesses** are selected that significantly contribute to overall process performance. These selected subprocesses are controlled using statistical and other quantitative techniques.
- **Quantitative objectives** for quality and process performance are established and used as criteria in managing processes. Quality and process performance are understood in statistical terms and are managed throughout the life of the processes.
- Special **causes of process variation** are identified and, where appropriate, the sources of special causes are corrected to prevent future occurrences.
- Quality and process performance measures are incorporated into the organization's measurement repository to support fact-based decision making in the future.

CMMI Staged Model - Maturity Levels 4 Quantitatively Managed

6.4 The CMMI Process Improvement Framework

Difference between Level 3 and 4:

- At level 4, the performance of processes is controlled using statistical and other quantitative techniques, and is **quantitatively** predictable. At level 3, processes are only **qualitatively** predictable.

CMMI Staged Model - Maturity Levels 5 Optimizing

6.4 The CMMI Process Improvement Framework

- Processes are continually improved based on quantitative understanding of the common causes of variation inherent in processes.
- Focuses on **continually improving process performance** through both incremental and innovative technological improvements.

- Difference between level 4 and 5:
  - ✓ At level 4, processes are concerned with **addressing special causes** of **process variation** and providing statistical predictability of the results. Though processes may produce predictable results, the results may be insufficient to achieve the established objectives.
  - ✓ At level 5, processes are concerned with addressing common causes of process variation and changing the process to improve process performance in order to achieve the established quantitative process-improvement objectives.

CMMI Staged Model – Process Areas

6.4 The CMMI Process Improvement Framework

Each CMM maturity level (ML) is accompanied by **Key Process Areas (KPA)** which must be met before promoting to the next higher level.

To move from level 1 to level 2, a company must develop competencies in these process areas

Level	Focus	Key Process Area	Result
1 Initial	Process is informal & Adhoc		Lowest quality / Highest risk
2 Managed	Basic Project Management	• Requirements Management • Project Planning • Project Monitoring & Control • Measurement & Analysis • Process & Product Quality Assurance • Supplier Agreement Management Configuration Management	Low quality / High Risk

CMMI Staged Model – Process Areas

6.4 The CMMI Process Improvement Framework

To move from level 2 to 3, competencies in these key process areas also must be developed

Level	Focus	Key Process Area	Result	
3 Defined	Process Standardization	• Requirements Development • Technical Solution • Product integration • Verification • Validation • Organizational Process Focus • Organizational Process Definition • Organizati…raining	• Integrated Project Management • Risk… • Dec… Res… • Inte… • Org… Int… Mar…	Medium quality / Medium risk
4 Quantitatively Managed	Quantitatively Managed	• Organizational process performance • Quantitative project management		Higher quality / Lower risk
5 Optimizing	Continuous Process Improvement	• Organizational innovation & deployment • Causal analysis & resolution		Highest quality / Lowest risk

Company at level 4 have achieved competencies in all the required process areas

At level 4, process performance are measured and controlled

CMMI Continuous Model

6.4 The CMMI Process Improvement Framework

- The continuous CMMI model does not classify an organization according to discrete levels. Rather, they are finer-grained models that consider individual or groups of practices and assess the use of good practices within each group process.
- It is a well-defined evolutionary describing the organization's capability relative to a process area.
- A capability level consists of related specific and generic practices for a process area that can improve the organization's processes associated with that process area. Each level is a layer in the foundation for continuous process improvement.
  - Thus, capability levels are cumulative, i.e., a higher capability level includes the attributes of the lower levels.

CMMI Continuous Model - Organization of Process Areas

6.4 The CMMI Process Improvement Framework

Category	Process Area
Project Management	• Project Planning • Project Monitoring and Control • Supplier Agreement Management • Integrated Project Management (IPPD) • Integrated Supplier Management (SS) • Integrated Teaming (IPPD) • Risk Management Quantitative Project Management
Support	• Configuration Management • Process and Product Quality Assurance • Measurement and Analysis Causal Analysis and Resolution • Decision Analysis and Resolution • Organizational Environment for Integration (IPPD)
Engineering	• Requirements Management • Requirements Development • Technical Solution • Product Integration • Verification • Validation
Process Management	• Organizational Process Focus • Organizational Process Definition

CMMI Continuous Model – Capability Level

6.4 The CMMI Process Improvement Framework

- The continuous CMMI assigns a capability level designated by the numbers 0 through 5 to each process area.

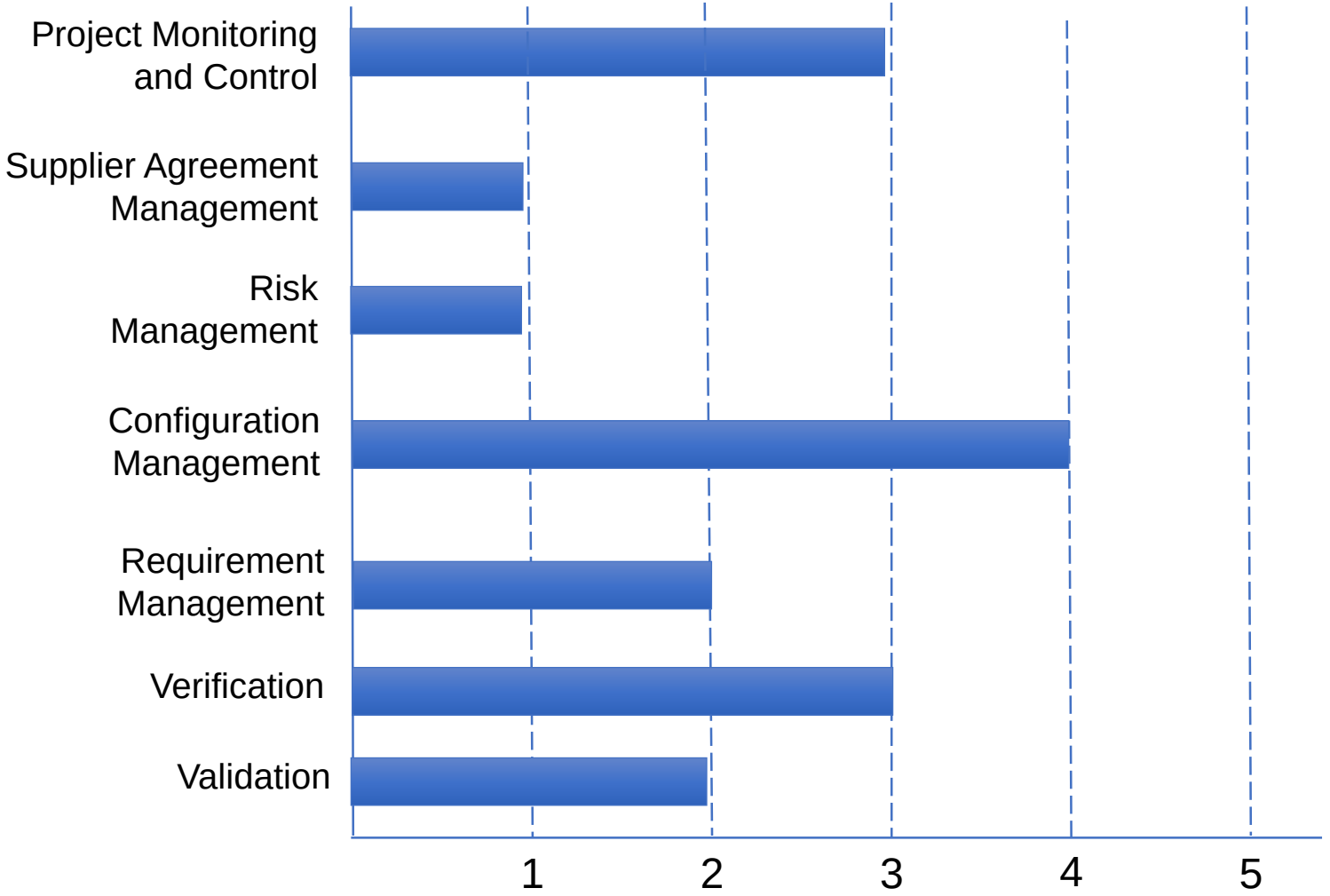
  - 0 − Incomplete
  - 1 − Performed
  - 2 − Managed
  - 3 − Defined
  - 4 − Quantitatively Managed
  - 5 − Optimizing

- Organizations operate at different capability levels for different process areas. Consequently, the result of a continuous CMMI assessment is a **capability profile** showing each process area and its associated capability level.

CMMI Continuous Model – Process Capability Profile

6.4 The CMMI Process Improvement Framework

Organizations can choose the process areas they want to focus on & when to improve them.



0 to 5 are level of capability

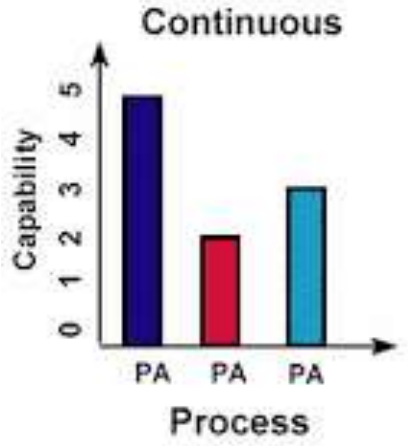
- 0 − Incomplete
- 1 − Performed
- 2 − Managed
- 3 − Defined
- 4 −Quantitatively Managed
- 5 − Optimizing

- In the process capability profile shown on the left, the capability level in
  - Configuration management is 4 (high)
  - Risk management is 1 (low)
- A company may develop actual and target capability profiles where the target profile reflects the capability level that they would like to reach for that process area.

CMMI Continuous vs Staged Model

6.4 The CMMI Process Improvement Framework

- CMMI Staged Model
  - Requires companies to focus on the different stages.
  - It defines a clear improvement pathway for organizations. They can plan to move from the second to the third level and so on.



- CMMI Continuous Model
  - Permits discretion and flexibility, while still allowing companies to work within the CMMI Improvement Framework.
  - Companies can pick and choose processes for improvement according to their own needs and requirements, e.g.
  - A company that develops software for the aerospace industry may focus on improvements in system specification, configuration management and validation
  - Eg: A web development company may be more concerned with customer-facing processes

Summary

6.1 The Rationale for Process Improvement

6.2 Software Process Improvement (SPI) Cycle

6.3 Process Exceptions

6.4 CMMI Process Improvement Framework