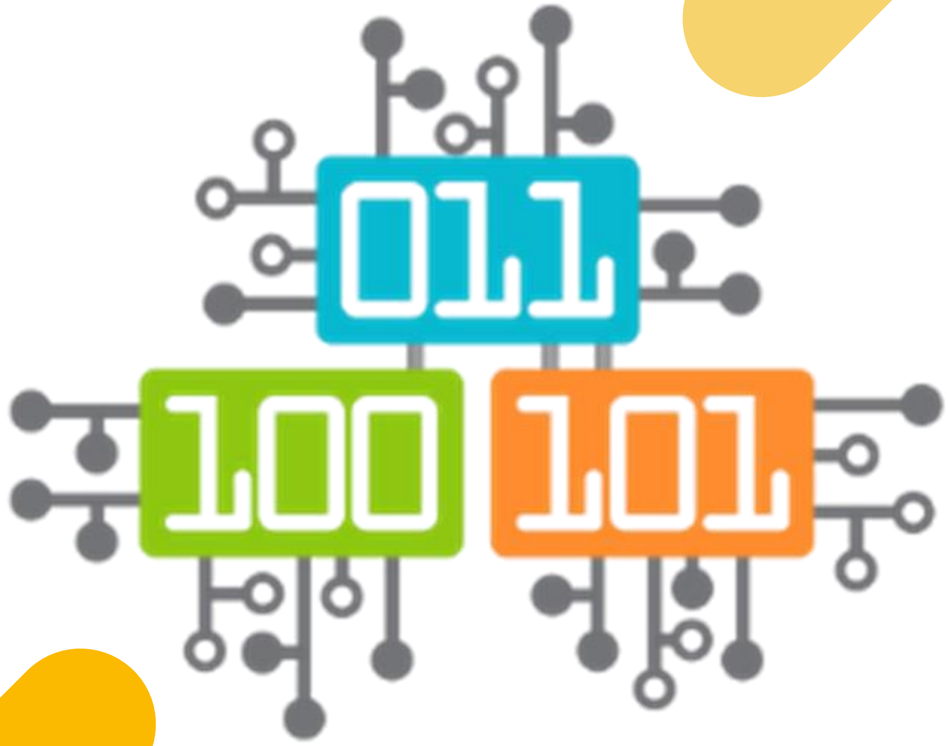


SOFTWARE ENGINEERING PROCESS

ALEXANDER BARÓN SALAZAR, PHD

SOFTWARE ENGINEERING PROCESS

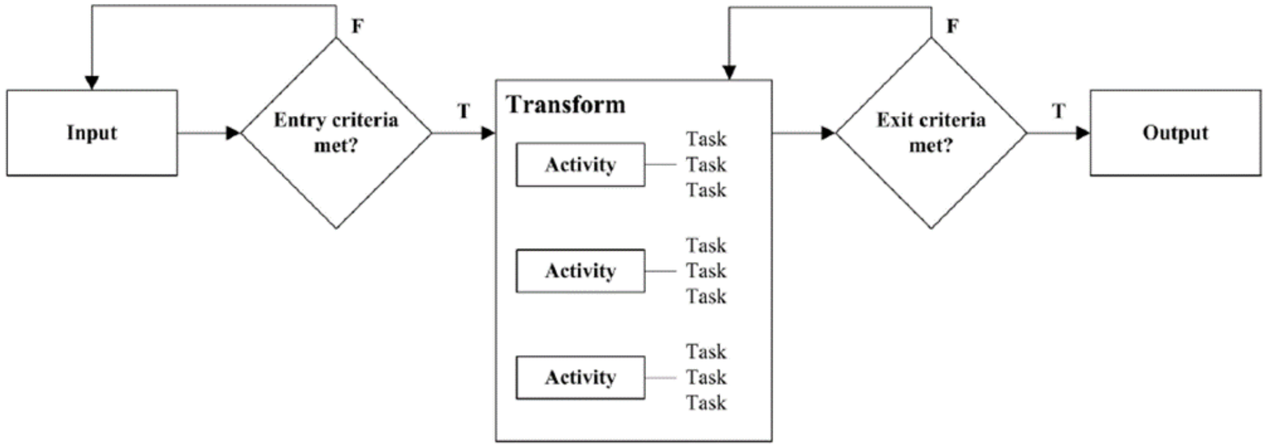
- **Software Process Definition**
- Software Life Cycles
- Software Process Assessment and Improvement

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS DEFINITION

An **engineering process** consists of a set of interrelated activities that transform one or more inputs into outputs while consuming resources to accomplish the transformation.

Software engineering processes are concerned with work activities accomplished by software engineers to develop, maintain, and operate software, such as requirements, design, construction, testing, configuration management, and other software engineering processes.

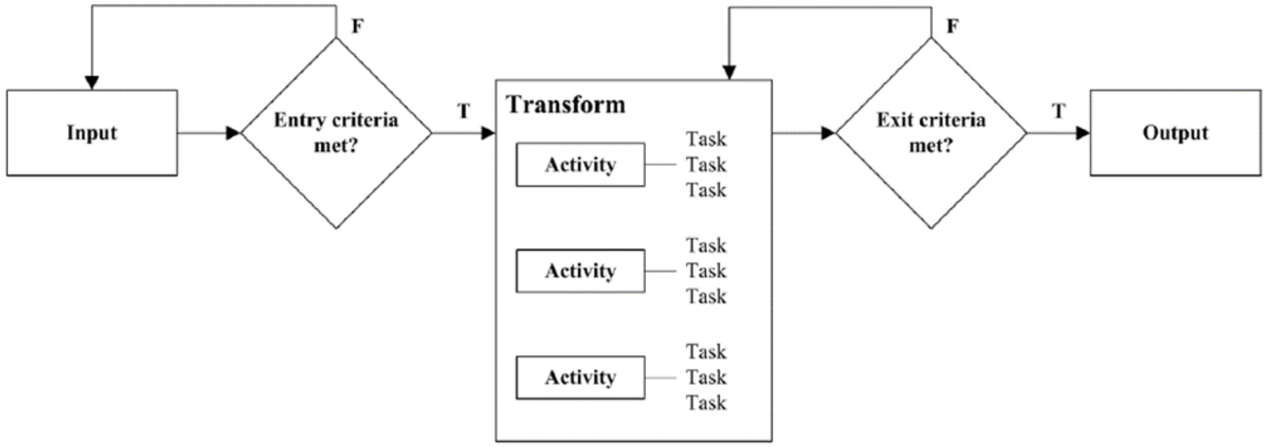
Software Engineering Process = Software Process



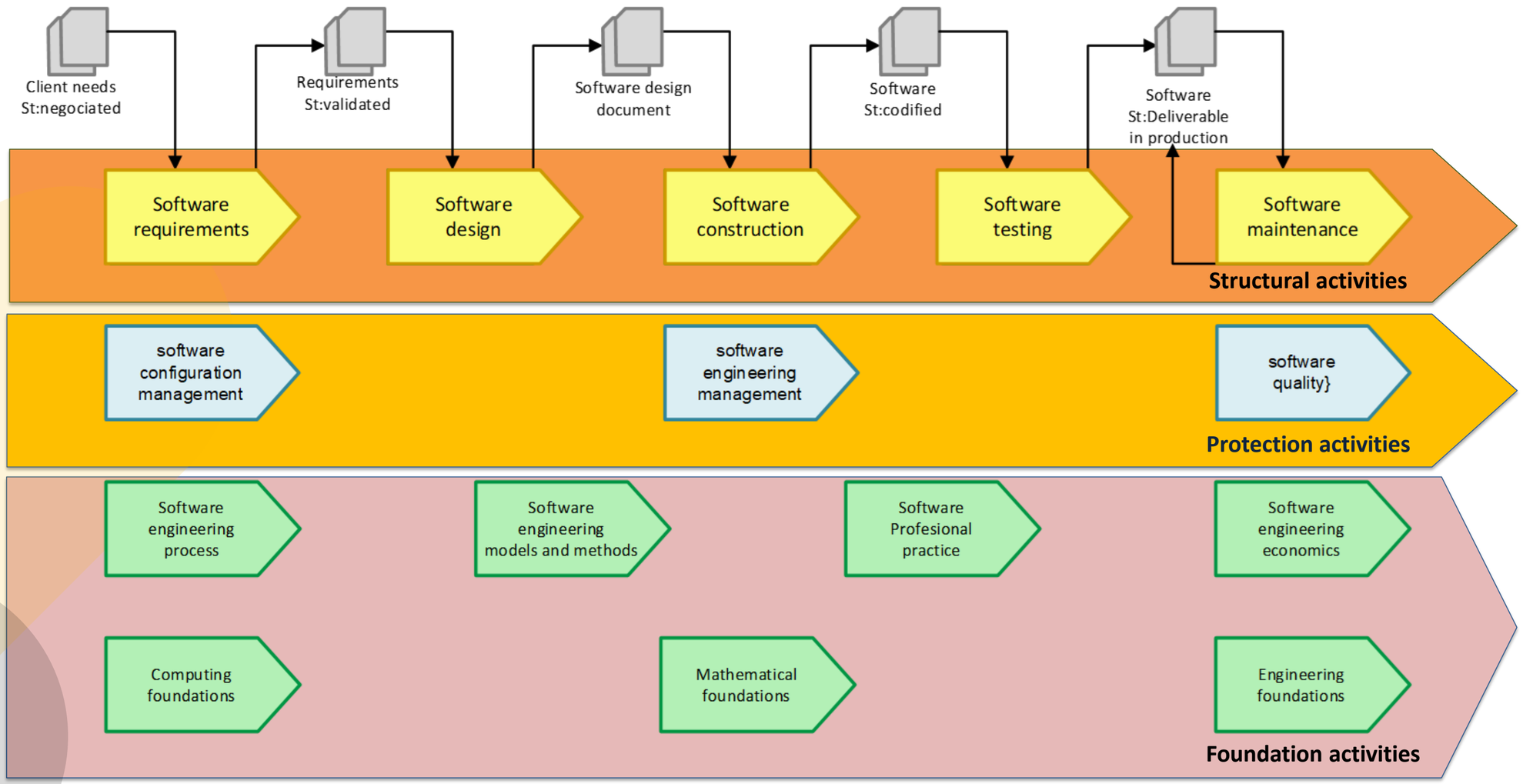
SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS DEFINITION

Software process is specified for:

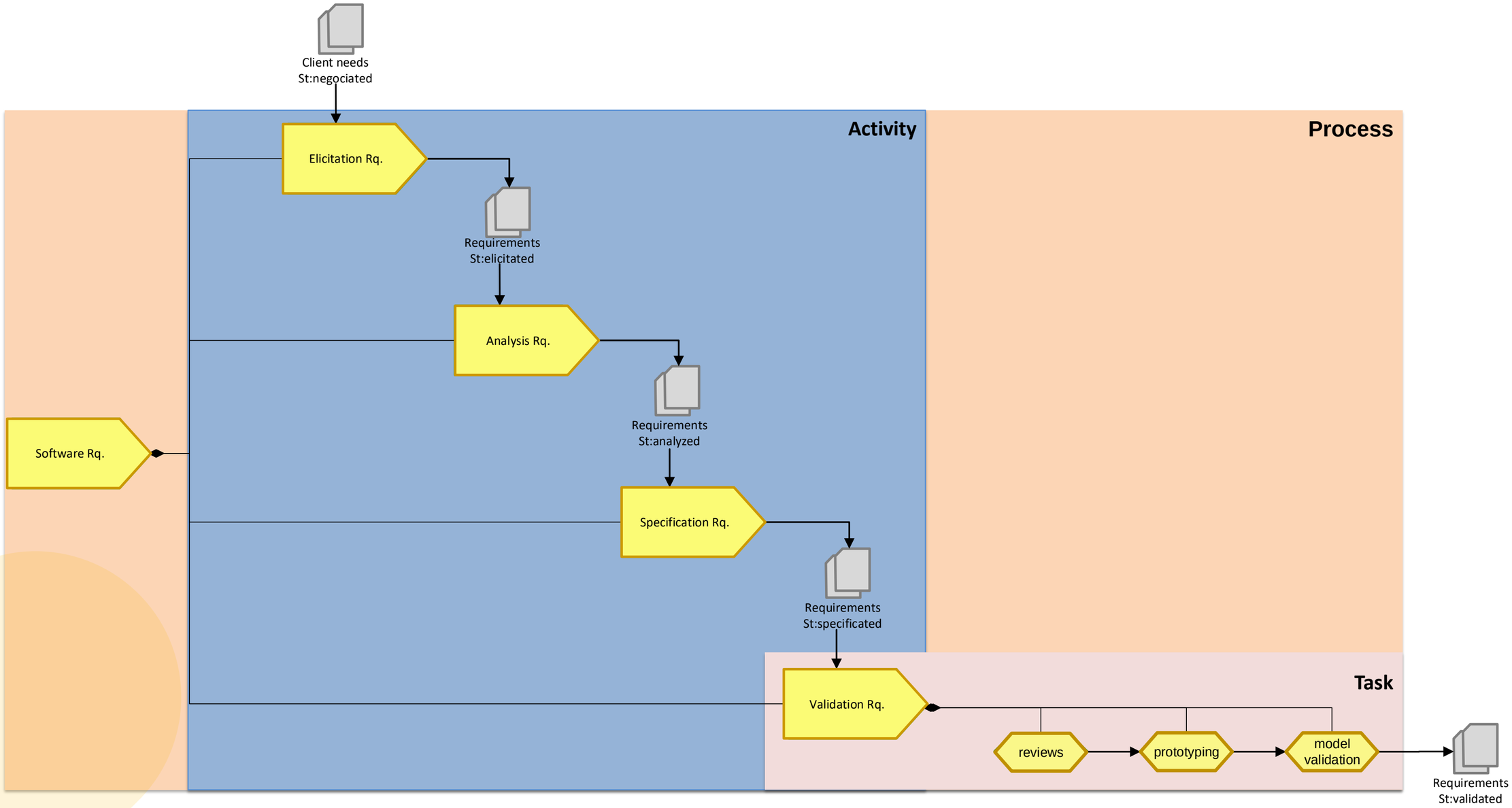
To facilitate human understanding, communication, and coordination.
To aid management of software projects
To measure and improve the quality of software products in an efficient manner
To support process improvement; and to provide a basis for automated support of process execution.



SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS DEFINITION



SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS DEFINITION



SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS DEFINITION

Other elements:

- Roles and competencies
- IT support,
- Software engineering techniques and tools,
- Work environment needed to perform the process
- Approaches and measures
- (Key Performance Indicators)

Other activities:

- Interleaved technical
- Collaborative
- Administrative

Notations:

- Textual lists
- Natural language
- Data-flow diagrams
- State charts
- BPMN
- UML activity diagram
- ...

¡There is no best software process or set of software processes. Software processes must be selected, adapted, and applied as appropriate for each project and each organizational context. No ideal process, or set of processes, exists!

SOFTWARE ENGINEERING PROCESS

- Software engineering processes
- Software Process Management
- Software Process Infrastructure

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS MANAGEMENT

Objectives:
- Realize the efficiency and effectiveness that result from a systematic approach to accomplishing software processes.
- Producing work products—be it at the individual, project, or organizational level
- Introduce new or improved processes.

Actions:
- Definition of a new process
- Improvement of an existing process
- Evolution of an existing process

Impacts:
- Improvement efficiency and/or effectiveness of the process
- Improvement the quality of the resulting work products.
- Improvement the cost, development schedule, and quality of the software products.

SOFTWARE ENGINEERING PROCESS

- Software engineering processes
- Software Process Management
- Software Process Infrastructure

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS INFRAESTRUCTURE

Establishing, implementing, and managing software processes

Individual software project level

Systematic application of software processes

Organizational project level

Software process infrastructure can provide:

- Process definitions
- Policies for interpreting and applying the processes
- Descriptions of the procedures to be used to implement the processes
- Funding, tools, training, and staff members who have been assigned responsibilities for establishing and maintaining the software process infrastructure

Software process infrastructure varies, depending on the size and complexity of the organization and the projects undertaken within the organization.

Myth

Establishing a software process infrastructure and implementing repeatable software processes will add time and cost to software development and maintenance.

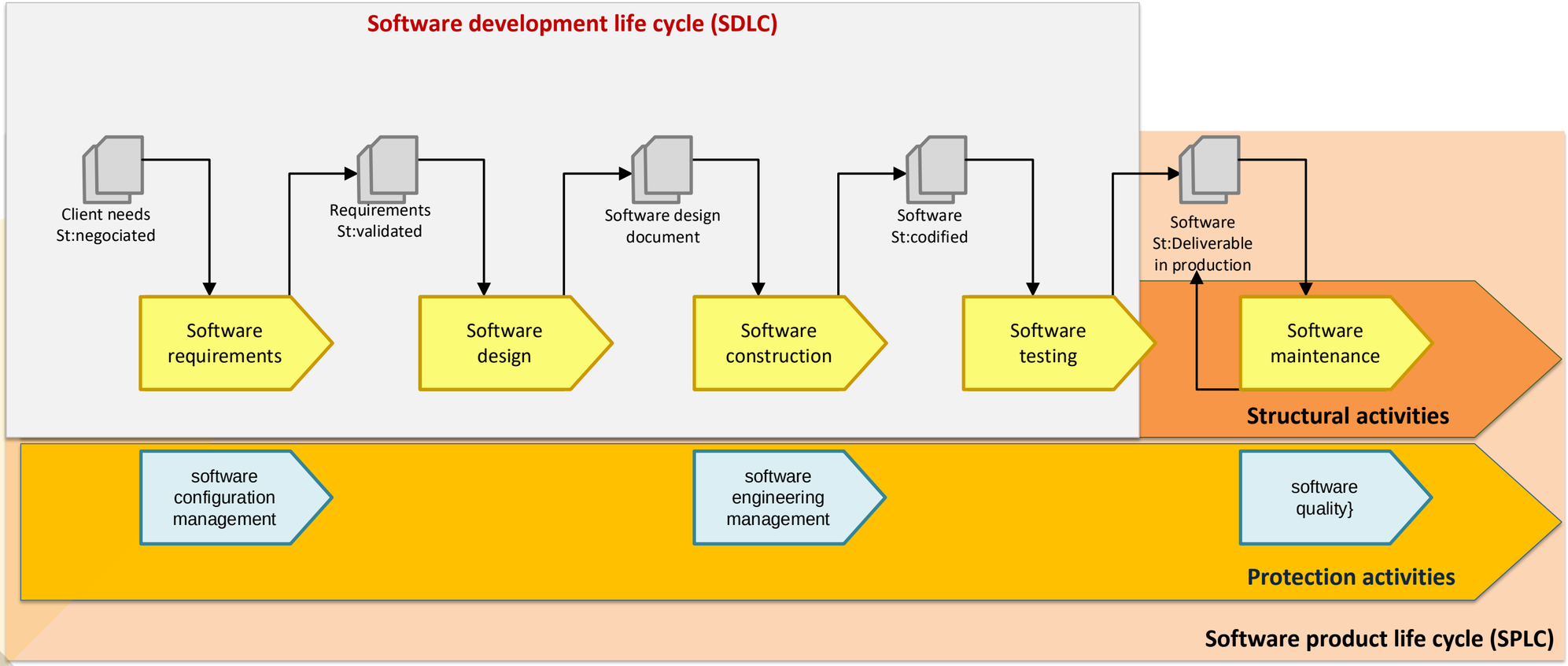
Reality

Implementing systematic improvement of software processes tends to result in lower cost through improved efficiency, avoidance of rework. Process performance thus influences software product quality.

SOFTWARE ENGINEERING PROCESS

- Software Process Definition
- **Software Life Cycles**
- Software Process Assessment and Improvement

SOFTWARE ENGINEERING PROCESS – SOFTWARE LIFE CYCLES



SOFTWARE ENGINEERING PROCESS

- Definitions
- Categories of Software Processes
- Software Life Cycle Models
- Software Process Adaptation
- Practical Considerations

SOFTWARE ENGINEERING PROCESS – CATEGORIES OF SOFTWARE PROCESSES

Many distinct software processes have been defined for use in the various parts of the software development and software maintenance life cycles. These processes can be categorized as follows:

1. Primary processes include software processes for development, operation, and maintenance of software.

2. Supporting processes are applied intermittently or continuously throughout a software product life cycle to support primary processes; they include software processes such as configuration management, quality assurance, and verification and validation.

3. Organizational processes provide support for software engineering; they include training, process measurement analysis, infrastructure management, portfolio and reuse management, organizational process improvement, and management of software life cycle models.

4. Cross-project processes, such as reuse, software product line, and domain engineering; they involve more than a single software project in an organization.

Other processes
Project management processes
Risk management processes
Software quality

SOFTWARE ENGINEERING PROCESS

- Definitions
- Categories of Software Processes
- Software Life Cycle Models
- Software Process Adaptation
- Practical Considerations

SOFTWARE ENGINEERING PROCESS – SOFTWARE LIFE CYCLES

The intangible and malleable nature of software permits a wide variety of software development life cycle models:

Predictive software development life cycle models

Lineal models
Phases of software development are accomplished sequentially with feedback and iteration as needed followed by integration, testing, and delivery of a single product;

Adaptive software development life cycle models

Iterative models in which software is developed in increments of increasing functionality on iterative cycles

Agile models that typically involve frequent demonstrations of working software to a customer or user representative who directs development of the software in short iterative cycles that produce small increments of working, deliverable software. Incremental, iterative, and agile models can deliver early subsets of working software into the user environment, if desired.

It should be noted that various maintenance activities during an SPLC can be conducted using different SDLC models, as appropriate to the maintenance activities.

SOFTWARE ENGINEERING PROCESS – SOFTWARE LIFE CYCLES

Predictive software development life cycle models

Develop a complete set of software requirements.

Adaptive software development life cycle models

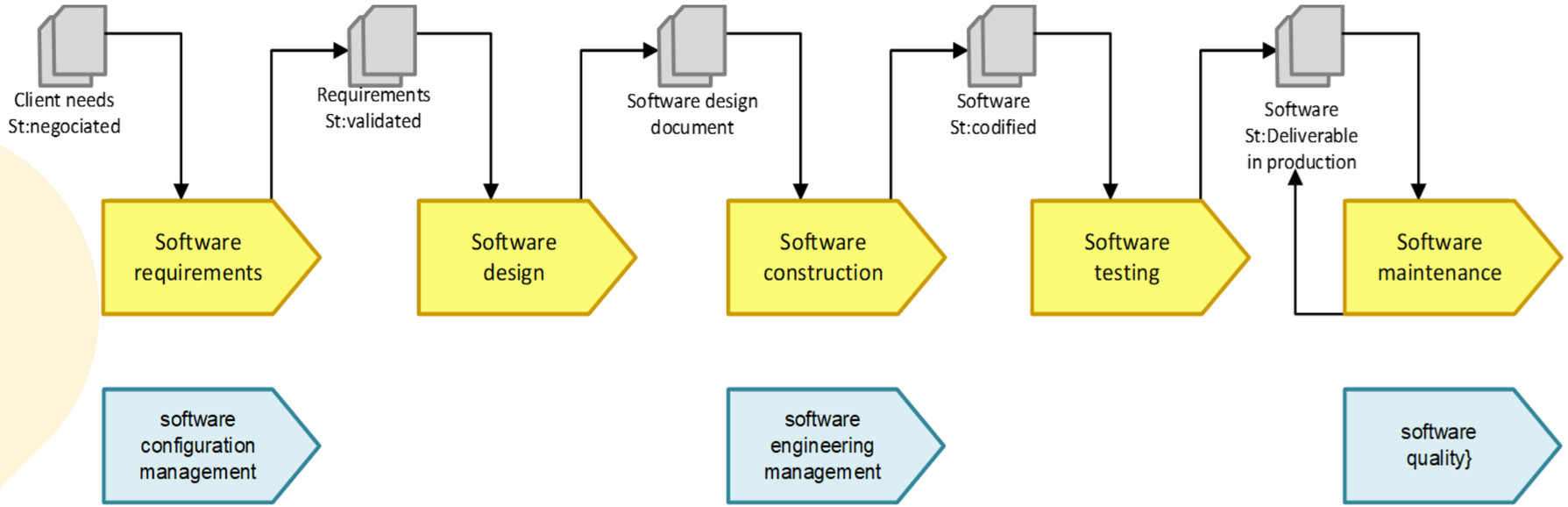
Deliverable software based on partitioning of the software requirements to be implemented in each of the increments.

Agile models are designed to facilitate evolution of the software requirements during the project.

It should be noted that various maintenance activities during an SPLC can be conducted using different SDLC models, as appropriate to the maintenance activities.

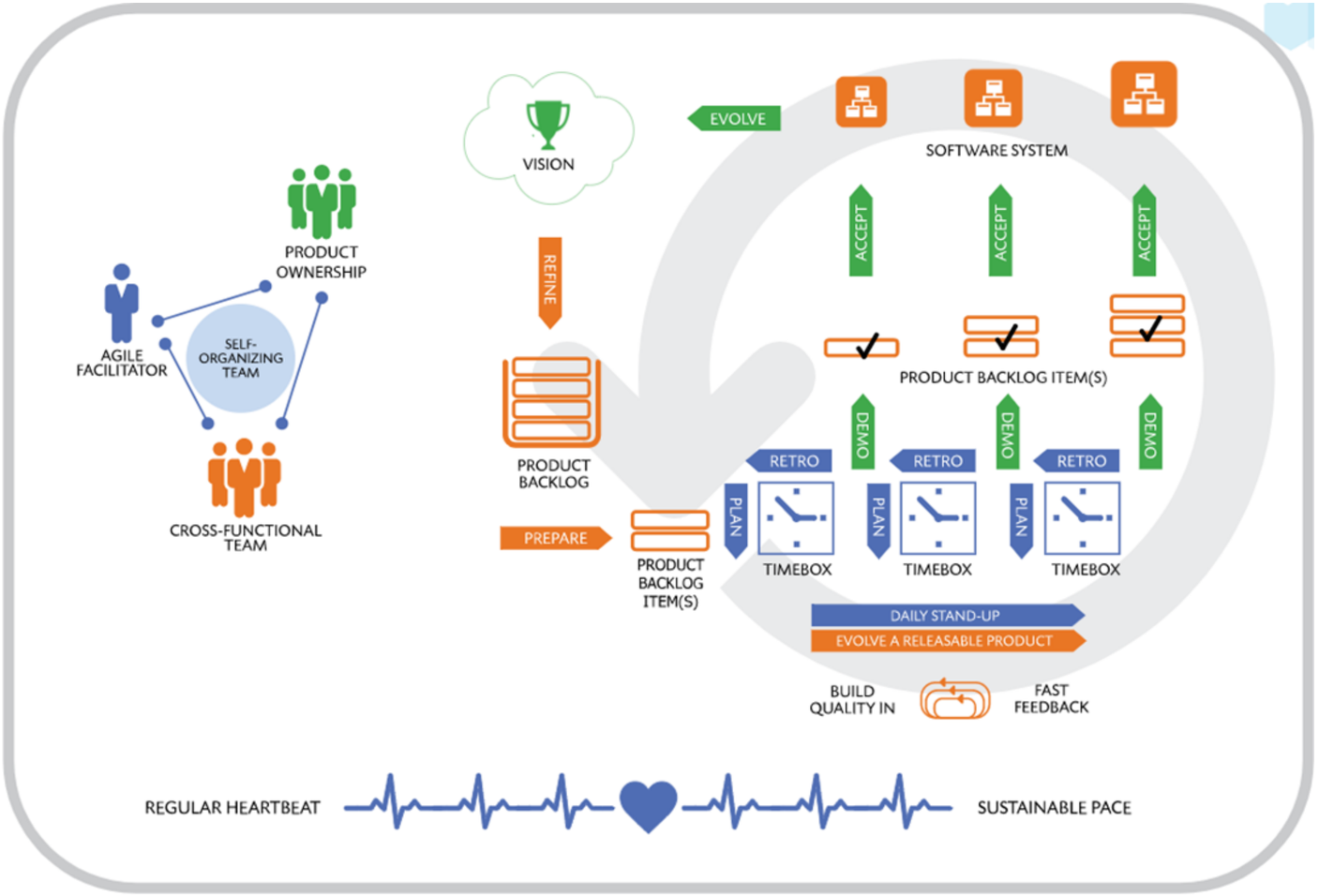
SOFTWARE ENGINEERING PROCESS – SOFTWARE LIFE CYCLES

Predictive software development life cycle models



SOFTWARE ENGINEERING PROCESS – SOFTWARE LIFE CYCLES

Adaptive software development life cycle models



SOFTWARE ENGINEERING PROCESS

- Definitions
- Categories of Software Processes
- Software Life Cycle Models
- Software Process Adaptation
- Practical Considerations

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS ADAPTATION

Organizational context, innovations in technology, project size, product criticality, regulatory requirements, industry practices, and corporate culture may determine needed adaptations.

Adding more details to software processes, activities, tasks, and procedures to address critical concerns.

Using an alternate set of activities that achieves the purpose and outcomes of the software process.

Mitting software processes or activities from a development or product life cycle model that are clearly inapplicable to the scope of work to be accomplished.

SOFTWARE ENGINEERING PROCESS

- Definitions
- Categories of Software Processes
- Software Life Cycle Models
- Software Process Adaptation
- Practical Considerations

SOFTWARE ENGINEERING PROCESS – PRACTICAL CONSIDERATIONS

Software processes and activities are often interleaved, overlapped, and applied concurrently. Software life cycle models that specify discrete software processes, with rigorously specified entry and exit criteria and prescribed boundaries and interfaces, should be recognized as idealizations.

Software processes (such as configuration management, construction, and testing) can be adapted to facilitate operation, support, maintenance, migration, and retirement of the software.

Required conformance to standards, directives, and policies; customer demands; criticality of the software product; and organizational maturity and competencies.

Nature of the work (e.g., modification of existing software versus new development) and the application domain (e.g., aerospace versus hotel management).

SOFTWARE ENGINEERING PROCESS

- Software Process Definition
- Software Life Cycles
- **Software Process Assessment and Improvement**

SOFTWARE ENGINEERING PROCESS – SPAI

- Definitions
- Software Process Assessment Models and Methods
- Software Process Improvement Models and Methods

SOFTWARE ENGINEERING PROCESS – DEFINITIONS

Software process assessments, capability evaluation and performance appraisals

- **Software process assessments**: form and content of a software process, which may be specified by a standardized set of criteria.
- **Capability evaluations**: software processes used by a supplier (or potential supplier) are acceptable to the acquirer.
- **Performance appraisals:** determine whether a process (or processes) satisfies the criteria at a given level of process capability or maturity.

SOFTWARE ENGINEERING PROCESS – SPAI

- Definitions
- Software Process Assessment Models and Methods
- Software Process Improvement Models and Methods

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS ASSESSMENT MODELS AND METHODS

Software process assessment models include assessment criteria for software processes that are regarded as constituting good practices.

A software process assessment method can be qualitative or quantitative.

Qualitative assessments rely on the judgment of experts

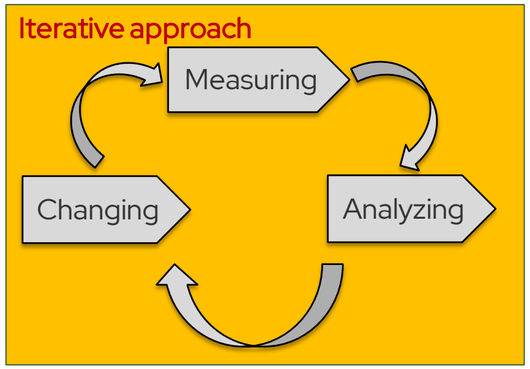
Quantitative assessments assign numerical scores to software processes based on analysis of objective evidence that indicates attainment of the goals and outcomes of a defined software process.

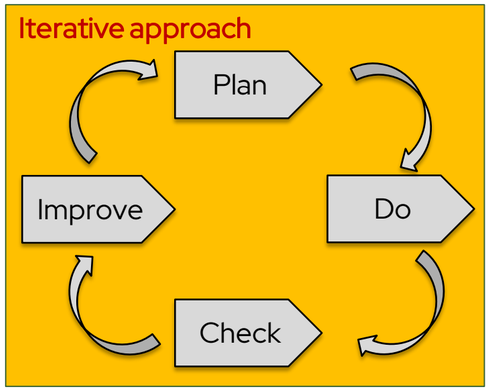
SOFTWARE ENGINEERING PROCESS – SPAI

- Definitions
- Software Process Assessment Models and Methods
- Software Process Improvement Models and Methods

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS IMPROVEMENT MODELS AND METHODS

Software process improvement emphasize iterative cycles of continuous improvement.





Improvement activities include identifying and prioritizing desired improvements (planning); introducing an improvement, including change management and training (doing); evaluating the improvement as compared to previous or exemplary process results and costs (checking); and making further modifications (acting).

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS IMPROVEMENT MODELS AND METHODS

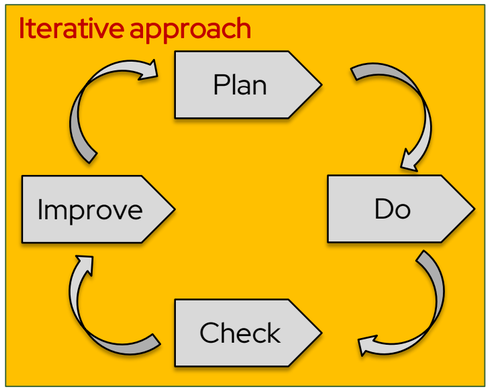
A continuous rating system involves assigning a rating to each software process of interest.

A staged rating system is stablished by assigning the same maturity rating to all of the software processes within a specified process level.

Level	Continuous Representation of Capability Levels	Staged Representation of Maturity Levels
0	Incomplete	
1	Performed	Initial
2	Managed	Managed
3	Defined	Defined
4		Quantitatively Managed
5		Optimizing

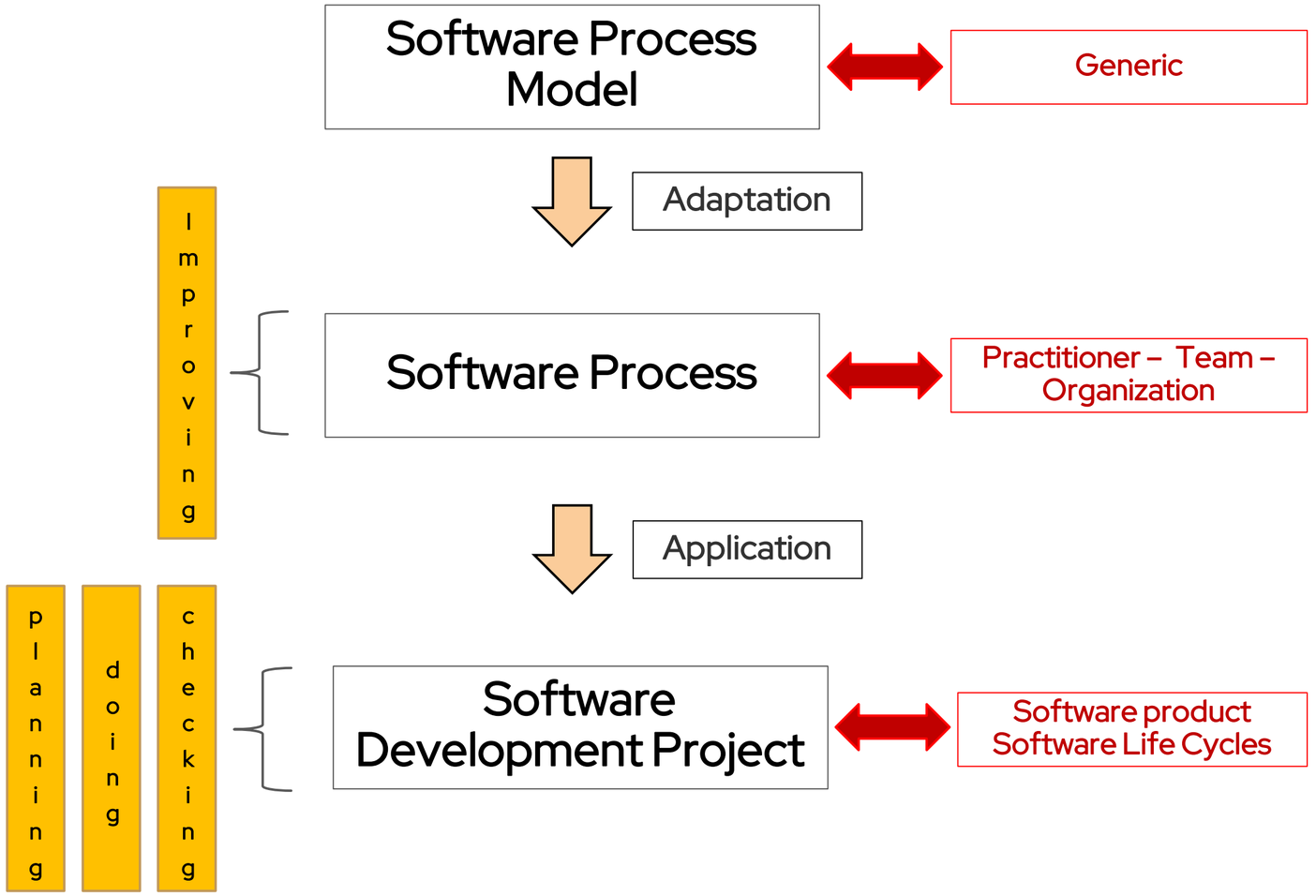
SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS IMPROVEMENT MODELS AND METHODS

Software process improvement emphasize iterative cycles of continuous improvement.



Improvement activities include identifying and prioritizing desired improvements (planning); introducing an improvement, including change management and training (doing); evaluating the improvement as compared to previous or exemplary process results and costs (checking); and making further modifications (acting).

SOFTWARE ENGINEERING PROCESS – SOFTWARE PROCESS ASSESSMENT AND IMPROVEMENT



THANKS!

ALEXANDER BARÓN SALAZAR, PHD