

Chapter 2

Process: A Generic View

Software Engineering: A Practitioner's Approach, 6th edition
by Roger S. Pressman



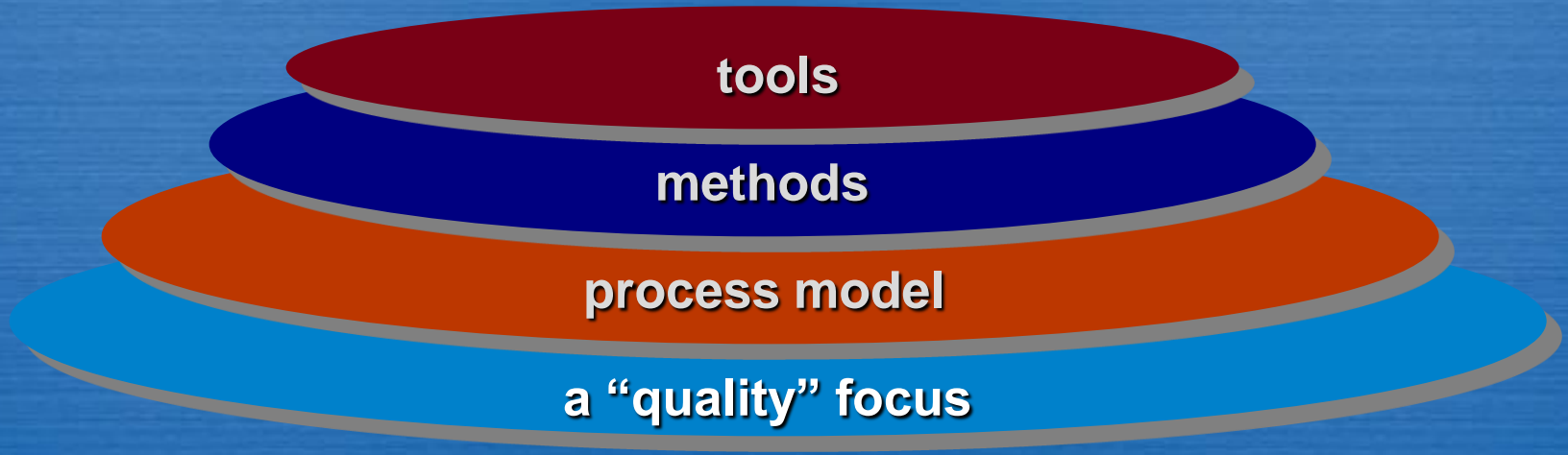
Chapter Overview

- **What?** A software process - a series of predictable steps that leads to a timely, high-quality product.
- **Who?** Managers, software engineers, and customers.
- **Why?** Provides stability, control, and organization to an otherwise chaotic activity.
- **Steps?** A handful of activities are common to all software processes, details vary.
- **Work product?** Programs, documents, and data.
- **Correct process?** Assessment, quality deliverable.



A Layered Technology

Software Engineering

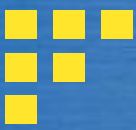




Software Engineering

Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

- IEEE Standard 610.12-1990



Software process **A Process Framework**

Process framework

Umbrella activities

framework activity #1

SE action #1.1

task set	{	work tasks work products QA points milestones
----------	---	--

SE action #1.2

task set	{	work tasks work products QA points milestones
----------	---	--

framework activity #2

SE action #2.1

task set	{	work tasks work products QA points milestones
----------	---	--

SE action #2.2

task set	{	work tasks work products QA points milestones
----------	---	--



Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management



Framework Activities

- **Communication**
- **Planning**
- **Modeling**
 - Analysis of requirements
 - Design
- **Construction**
 - Code generation
 - Testing
- **Deployment**




The Process Model: Adaptability

- The framework activities will always be applied on every project ... BUT
- The tasks (and degree of rigor) for each activity will vary based on:
 - the type of project
 - characteristics of the project
 - common sense judgment; concurrence of the project team



The CMMI

- The CMMI defines each process area in terms of “specific goals” and the “specific practices” required to achieve these goals.
- *Specific goals* establish the characteristics that must exist if the activities implied by a process area are to be effective.
- *Specific practices* refine a goal into a set of process-related activities.



Personal Software Process (PSP)

- Recommends five framework activities:

Planning

High-level design

High-level design review

Development

Postmortem

- Stresses the need for each software engineer to identify errors early and as important, to understand the types of errors



Team Software Process (TSP)

- Each project is “launched” using a “script” that defines the tasks to be accomplished
- Teams (of 2 to 20 engineers) are self-directed:
 - Plan and track work, set goals, own processes and plans
- Measurement is encouraged
- Measures are analyzed with the intent of improving the team process (through coaching, motivation, ...)



Process Patterns

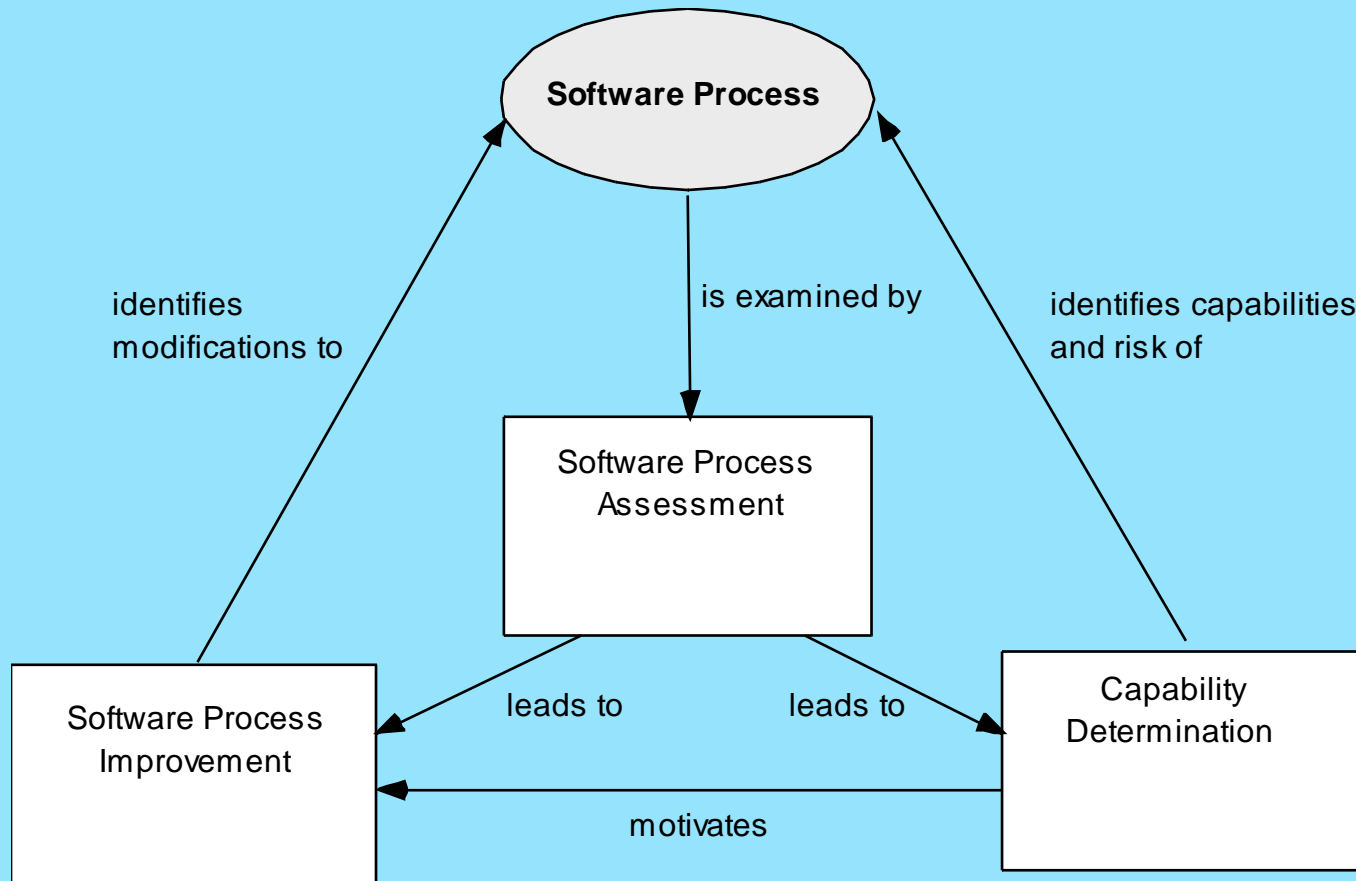
- Process patterns define a set of activities, actions, work tasks, work products and/or related behaviors
- A template is used to define a pattern
- Typical examples:
 - Customer communication (a process activity)
 - Analysis (an action)
 - Requirements gathering (a process task)
 - Reviewing a work product (a process task)
 - Design model (a work product)



Process Assessment

- The process should be assessed to ensure that it meets a set of basic process criteria that have been shown to be essential for a successful software engineering.
- Many different assessment options are available:
 - SCAMPI
 - CBA IPI
 - SPICE
 - ISO 9001:2000

Assessment and Improvement





The Primary Goal of Any Software Process: *High Quality*

Remember:

High quality \Rightarrow project timeliness

Why?

Less rework!