

# Chapter 30

---

## ■ **Software Process Improvement**

*Slide Set to accompany*

*Software Engineering: A Practitioner's Approach, 7/e*  
by **Roger S. Pressman**

Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

***For non-profit educational use only***

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

# What is SPI?

---

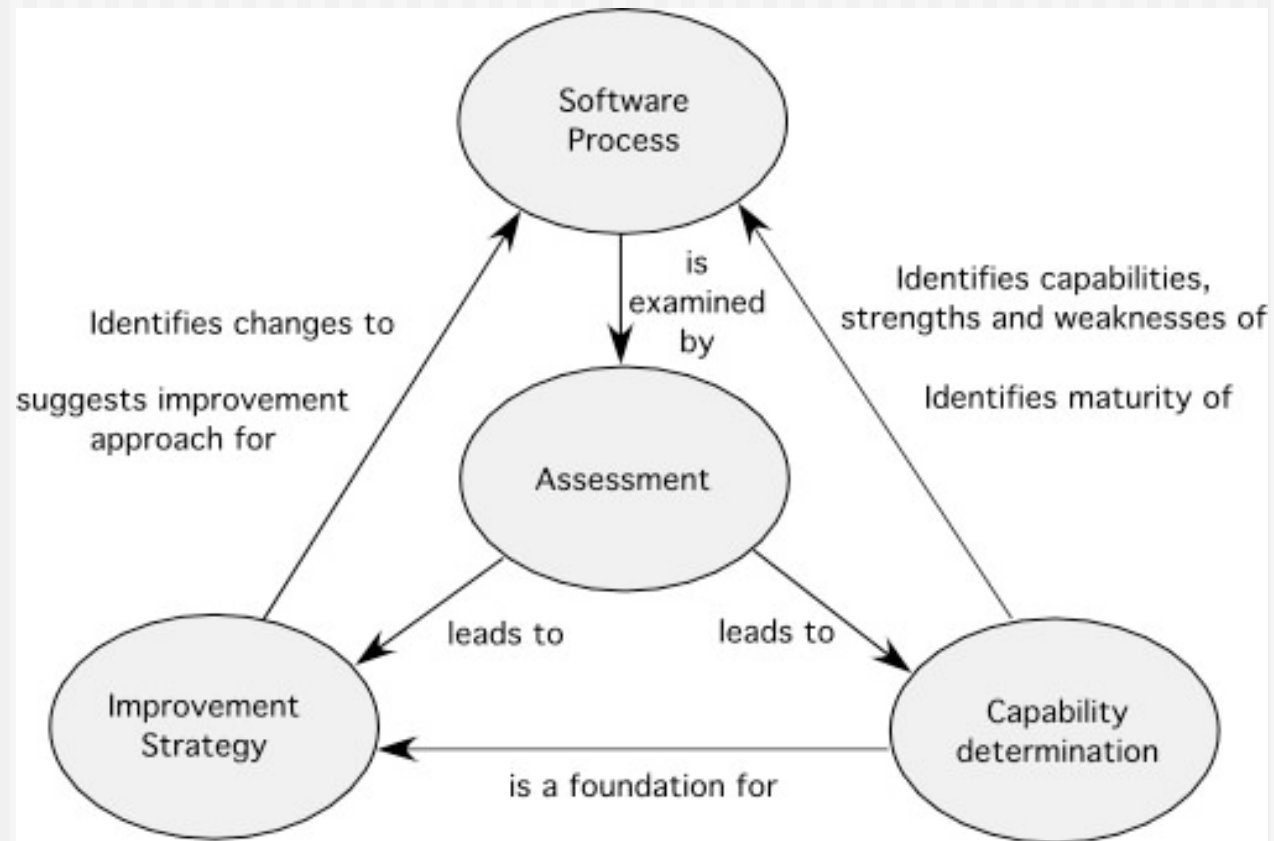
- SPI implies that
  - elements of an effective software process can be defined in an effective manner
  - an existing organizational approach to software development can be assessed against those elements, and
  - a meaningful strategy for improvement can be defined.
- The SPI strategy transforms the existing approach to software development into something that is **more focused, more repeatable, and more reliable** (in terms of the quality of the product produced and the timeliness of delivery).

# SPI Framework

---

- a **set of characteristics** that must be present if an effective software process is to be achieved
- a **method for assessing** whether those characteristics are present
- a **mechanism for summarizing the results** of any assessment, and
- a **strategy for assisting** a software organization in implementing those process characteristics that have been found to be weak or missing.
- An SPI framework assesses the **“maturity”** of an organization’s software process and provides a qualitative indication of a maturity level.

# Elements of a SPI Framework



These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/e* (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

# Constituencies

---

- Quality certifiers
  - Quality(Process) --> Quality(Product)
- Formalists
- Tool advocates
- Practitioners
- Reformers
- Ideologists

# Maturity Models

---

- A *maturity model* is applied within the context of an SPI framework.
- The intent of the maturity model is to provide an overall indication of the “process maturity” exhibited by a software organization.
  - an indication of the quality of the software process, the degree to which practitioner’s understand and apply the process,
  - the general state of software engineering practice.

# Is SPI for Everyone?

---

- Can a small company initiate SPI activities and do it successfully?
  - Answer: a qualified “yes”
- It should come as no surprise that small organizations are more informal, apply fewer standard practices, and tend to be self-organizing.
  - SPI will be approved and implemented only after its proponents demonstrate *financial leverage*.

# The SPI Process—I

---

## ■ Assessment and Gap Analysis

- *Assessment* examines a wide range of actions and tasks that will lead to a high quality process.
  - **Consistency.** Are important activities, actions and tasks applied consistently across all software projects and by all software teams?
  - **Sophistication.** Are management and technical actions performed with a level of sophistication that implies a thorough understanding of best practice?
  - **Acceptance.** Is the software process and software engineering practice widely accepted by management and technical staff?
  - **Commitment.** Has management committed the resources required to achieve consistency, sophistication and acceptance?
- *Gap analysis*—The difference between local application and best practice represents a “gap” that offers opportunities for improvement.



# The SPI Process—II

---

- *Education and Training*
- Three types of education and training should be conducted:
  - **Generic concepts and methods.** Directed toward both managers and practitioners, this category stresses both process and practice. The intent is to provide professionals with the intellectual tools they need to apply the software process effectively and to make rational decisions about improvements to the process.
  - **Specific technology and tools.** Directed primarily toward practitioners, this category stresses technologies and tools that have been adopted for local use. For example, if UML has been chosen for analysis and design modeling, a training curriculum for software engineering using UML would be established.
  - **Business communication and quality-related topics.** Directed toward all stakeholders, this category focuses on “soft” topics that help enable better communication among stakeholders and foster a greater quality focus.

# The SPI Process—III

---

- **Selection and Justification**
  - choose the process model (Chapters 2 and 3) that best fits your organization, its stakeholders, and the software that you build
  - decide on the set of framework activities that will be applied, the major work products that will be produced and the quality assurance checkpoints that will enable your team to assess progress
  - develop a work breakdown for each framework activity (e.g., modeling), defining the task set that would be applied for a typical project
  - Once a choice is made, time and money must be expended to install it within an organization and these resource expenditures should be justified.

# The SPI Process—IV

---

## ■ Installation/Migration

- actually *software process redesign* (SPR) activities. Scacchi [Sca00] states that “SPR is concerned with identification, application, and refinement of new ways to dramatically improve and transform software processes.”
- three different process models are considered:
  - the existing (“as-is”) process,
  - a transitional (“here-to-there”) process, and
  - the target (“to be”) process.

# The SPI Process—V

---

- **Evaluation**
  - assesses the degree to which changes have been instantiated and adopted,
  - the degree to which such changes result in better software quality or other tangible process benefits, and
  - the overall status of the process and the organizational culture as SPI activities proceed
- From a qualitative point of view, past management and practitioner attitudes about the software process can be compared to attitudes polled after installation of process changes.

# Risk Management for SPI

---

- manage risk at three key points in the SPI process [Sta97b]:
  - prior to the initiation of the SPI roadmap,
  - during the execution of SPI activities (assessment, education, selection, installation), and
  - during the evaluation activity that follows the instantiation of some process characteristic.
- In general, the following categories [Sta97b] can be identified for SPI risk factors:
  - budget and cost
  - content and deliverables culture
  - maintenance of SPI deliverables
  - mission and goals
  - organizational management and organizational stability
  - process stakeholders
  - schedule for SPI development
  - SPI development environment and process
  - SPI project management and SPI staff

# Critical Success Factors

---

- The top five CSFs are [Ste99]:
  - **Management commitment and support**
  - **Staff involvement**
  - **Process integration and understanding**
  - **A customized SPI strategy**
  - **Solid Management of the SPI project**

# The CMMI

---

- a comprehensive process meta-model that is predicated on a set of system and software engineering capabilities that should be present as organizations reach different levels of process capability and maturity
- a process meta-model in two different ways: (1) as a “continuous” model and (2) as a “staged” model
- 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.

# The People CMM

---

- “a roadmap for implementing workforce practices that continuously improve the capability of an organization’s workforce.” [Cur02]
- defines a set of five organizational maturity levels that provide an indication of the relative sophistication of workforce practices and processes



# P-CMM Process Areas

Level	Focus	Process Areas
Optimized	<i>Continuous improvement</i>	Continuous workforce innovation Organizational performance alignment Continuous capability improvement
Predictable	<i>Quantifies and manages knowledge, skills and abilities</i>	Mentoring Organizational capability management Quantitative performance management Competency-based assets Empowered workgroups Competency integration
Defined	<i>Identifies and develops knowledge, skills and abilities</i>	Participatory culture Workgroup development Competency-based practices Career development Competency development Workforce planning Competency analysis
Managed	<i>Repeatable, basic people management practices</i>	Compensation Training and development Performance management Work environment Communication and coordination Staffing
Initial	<i>Inconsistent practices</i>	

These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

# Other SPI Frameworks

---

- **SPICE**— a international initiative to support the International Standard ISO/IEC 15504 for (Software) Process Assessment [ISO08]
- **Bootstrap**—a SPI framework for small and medium sized organizations that conforms to SPICE [Boo06],
- **PSP and TSP**—individual and team specific SPI frameworks ([Hum97], [Hum00]) that focus on process in-the-small, a more rigorous approach to software development coupled with measurement
- **TickIT**—an auditing method [Tic05] that assesses an organization compliance to ISO Standard 9001:2000

# SPI Return on Investment

---

- “How do I know that we’ll achieve a reasonable return for the money we’re spending?”
  - $ROI = [\Sigma (\text{benefits}) - \Sigma (\text{costs})] / \Sigma (\text{costs}) \times 100\%$
- where
  - *benefits* include the cost savings associated with higher product quality (fewer defects), less rework, reduced effort associated with changes, and the income that accrues from shorter time-to-market.
  - *costs* include both direct SPI costs (e.g., training, measurement) and indirect costs associated with greater emphasis on quality control and change management activities and more rigorous application of software engineering methods (e.g., the creation of a design model).

# SPI Trends

---

- future SPI frameworks must become significantly more agile
- Rather than an organizational focus (that can take years to complete successfully), contemporary SPI efforts should focus on the project level
- To achieve meaningful results (even at the project level) in a short time frame, complex framework models may give way to simpler models.
- Rather than dozens of key practices and hundreds of supplementary practices, an agile SPI framework should emphasize only a few pivotal practices