Software process improvement

- Software process improvement encompasses a set of activities that will lead to a better software process and, as a consequence, higher-quality software delivered in a more timely manner. It implies many things:
- First, it implies that elements of an effective software process can be defined in more effective manner;
- Second, that an existing organizational approach to software development can be assessed against those elements;
- Third, that a meaningful strategy for improvement can be defined.

- Because SPI is not free, it must deliver a return on investment. The effort and time that is required to implement an SPI strategy must pay for itself in some measurable way.
- To do this, the results of improved process and practice must lead to:

1. a reduction in software “problems” that cost time and money.
2. It must reduce the number of defects that are delivered to end users,
3. reduce the amount of rework due to quality problems,
4. reduce the costs associated with software maintenance and support,
5. reduce the indirect costs that occur when software is delivered late.

Approaches to SPI

- An *SPI framework* defines

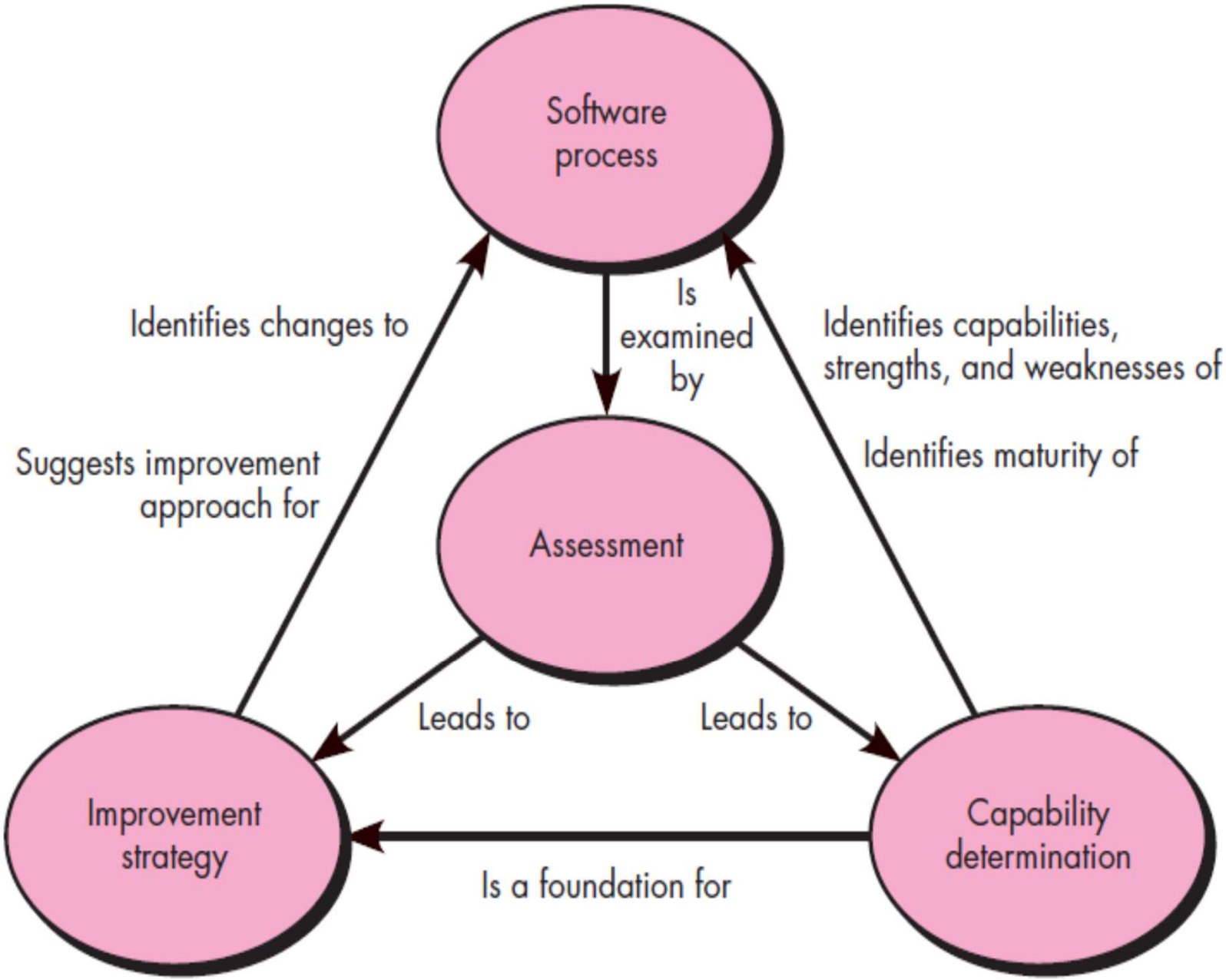
(1) a set of characteristics that must be present if an effective software process is to be achieved

(2) A method for assessing whether those characteristics are present.

(3) A mechanism for summarizing the results of any assessment.

(4) 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.
- the SPI framework encompasses a maturity model that in turn incorporates a set of process quality indicators that provide an overall measure of the process quality that will lead to product quality.
- Following figure provides an overview of a typical SPI framework.
- The key elements of the framework and their relationship to one another are shown.


Software process
Identifies changes to
Is examined by
Identifies capabilities, strengths, and weaknesses of
Suggests improvement approach for
Identifies maturity of
Assessment
Leads to
Leads to
Improvement strategy
Capability determination
Is a foundation for

The SPI framework that is chosen by an organization reflects the constituency that is championing the SPI effort. Following are defines six different SPI support constituencies:

1. Quality certifiers. Process improvement efforts championed by this group focus on the following relationship:

Quality(*Process*) ⇒ **Quality**(*Product*)

- Their approach is to emphasize assessment methods and to examine a well-defined set of characteristics that allow them to determine whether the process exhibits quality.
- They are most likely to adopt a process framework such as the CMM, SPICE, TickIT, or Bootstrap.

2. **Formalists.** This group wants to understand (and when possible, optimize) process workflow. To accomplish this, they use process modeling languages (PMLs) to create a model of the existing process and then design extensions or modifications that will make the process more effective.

3. Tool advocates. This group insists on a tool-assisted approach to SPI that models workflow and other process characteristics in a manner that can be analyzed for improvement.

- **Practitioners.** This constituency uses a pragmatic approach, “emphasizing mainstream project-, quality- and product management, applying project level planning and metrics, but with little formal process modeling or enactment support.
- **Reformers.** The goal of this group is organizational change that might lead to a better software process. They tend to focus more on human issues and emphasize measures of human capability and structure.
- **Ideologists.** This group focuses on the suitability of a particular process model for a specific application domain or organizational structure. Rather than typical software process models (e.g., iterative models), ideologists would have a greater interest in a process that would, say, support reuse or reengineering.
- As an SPI framework is applied, the sponsoring constituency (regardless of its overall focus) must establish mechanisms to:

(1) support technology transition

(2) determine the degree to which an organization is ready to absorb process changes that are proposed

(3) measure the degree to which changes have been adopted

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.
- That is, an indication of the quality of the software process, the degree to which practitioner's understand and apply the process, and the general state of software engineering practice.
- This is accomplished using some type of ordinal scale. For example, the Software Engineering Institute's *Capability Maturity Model* suggests five levels of maturity as follows:

- ***Level 1,*** Initial—Few processes are defined, and success depends more on individual heroic efforts than on following a process and using a synergistic team effort.
- ***Level 2, Repeatable***—Basic project management processes are established to track cost, schedule, and functionality. Planning and managing new products is based on experience with similar projects.
- ***Level 3, Defined***—Processes for management and engineering are documented, standardized, and integrated into a standard software process for the organization All projects use an approved, tailored version of the organization's standard software process for developing software.

Level 4, Managed—Detailed software process and product quality metrics establish the quantitative evaluation foundation. Meaningful variations in process performance can be distinguished from random noise, and trends in process and product qualities can be predicted.

Level 5, Optimized—The organization has quantitative feedback systems in place to identify process weaknesses and strengthen them pro-actively. Project teams analyze defects to determine their causes; software processes are evaluated and updated to prevent known types of defects from recurring.

THE SPI PROCESS

- The Software Engineering Institute has developed IDEAL—"an organizational improvement model that serves as a roadmap for initiating, planning, and implementing improvement actions".
- IDEAL is representative of many process models for SPI, defining five distinct activities—initiating, diagnosing, establishing, acting, and learning—that guide an organization through SPI activities.
- Based on the process model for SPI originally proposed in It applies a commonsense philosophy that requires an organization to

(1) look in the mirror,

(2) then get smarter so it can make intelligent choices,

(3) select the process model (and related technology elements) that best meets its needs, (

4) instantiate the model into its operating environment and its culture, and (5) evaluate what has been done.

These five activities are applied in an iterative (cyclical) manner in an effort to foster continuous process improvement.

1. Assessment and Gap Analysis

- The intent of assessment is to uncover both strengths and weaknesses in the way your organization applies the existing software process and the software engineering practices that populate the process.
- Assessment examines a wide range of actions and tasks that will lead to a high quality process.
- For example, regardless of the process model that is chosen, the software organization must establish generic mechanisms such as:

1. defined approaches for customer communication;
2. established methods for representing user requirements;
3. defining a project management framework that includes scoping, estimation,
4. scheduling,
5. project tracking;
6. risk analysis methods;
7. change management procedures;
8. quality assurance and control activities including reviews; and many others.

- As the process assessment is conducted, you (or those who have been hired to perform the assessment) should also focus on the following attributes:

1. ***Consistency***. Are important activities, actions, and tasks applied consistently across all software projects and by all software teams?
2. ***Sophistication***. Are management and technical actions performed with a level of sophistication that implies a thorough understanding of best practice?
3. ***Acceptance***. Is the software process and software engineering practice widely accepted by management and technical staff?
4. ***Commitment***. Has management committed the resources required to achieve consistency, sophistication, and acceptance?

- The difference between local application and best practice represents a "gap" that offers opportunities for improvement.
- The degree to which consistency, sophistication, acceptance, and commitment are achieved indicates the amount of cultural change that will be required to achieve meaningful improvement.

2. Education and Training

- A key element of any SPI strategy is education and training for practitioners, technical managers and more senior managers who have direct contact with the software organization. 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.

3. Selection and Justification

In selection and justification process characteristics and specific software engineering methods and tools are chosen to populate the software process.

- First, you should choose the process model that best fits your organization, its stakeholders, and the software that you build. You should decide which of the set of framework activities will be applied, the major work products that will be produced, and the quality assurance checkpoints that will enable your team to assess progress.
- Next, develop a work breakdown for each framework activity (e.g., modeling), defining the task set that would be applied for a typical project. You should also consider the software engineering methods that can be applied to achieve these tasks.
- You should also consider the software engineering methods that can be applied to achieve these tasks. In reality, selection can be a rocky road. It is often difficult to achieve consensus among different constituencies.
- It is true that a bad choice can do more harm than good, As long as the process characteristic or technology element has a good chance at meeting an organization's needs, it's sometimes better to pull the trigger and make a choice, rather than waiting for the optimal solution.
- Once a choice is made, time and money must be expended to instantiate it within an organization, and these resource expenditures should be justified

Installation/Migration

- *Installation* is the first point at which a software organization feels the effects of changes implemented as a consequence of the SPI road map.
- In some cases, an entirely new process is recommended for an organization. Framework activities, software engineering actions, and individual work tasks must be defined and installed as part of a new software engineering culture.
- Installation and migration are actually *software process redesign* (SPR) activities
- "SPR is concerned with identification, application, and refinement of new ways to dramatically improve and transform software processes."
- When a formal approach to SPR is initiated, three different process models are considered:

(1) the existing ("as is") process,

(2) a transitional ("here to there") process,

(3) the target ("to be") process.

- If the target process is significantly different from the existing process, the only rational approach to installation is an incremental strategy in which the transitional process is implemented in steps.

Evaluation

- *Evaluation* occurs throughout SPI. The evaluation activity 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.
- Both qualitative factors and quantitative metrics are considered during the evaluation activity.
- 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.
- Quantitative metrics are collected from projects that have used the transitional or “to be” process and compared with similar metrics that were collected for projects that were conducted under the “as is” process.

Risk Management for SPI

- SPI is a risky undertaking. In fact, more than half of all SPI efforts end in failure.
- Among the most common risks are:

1. a lack of management support
2. cultural resistance by technical staff,
3. a poorly planned SPI strategy,
4. an overly formal approach to SPI,
5. selection of an inappropriate process,
6. a lack of buy-in by key stakeholders,
7. an inadequate budget,
8. a lack of staff training,
9. organizational instability etc.

- A software organization should manage risk at three key points in the SPI process:

1. prior to the initiation of the SPI road map,
2. during the execution of SPI activities (assessment, education, selection, installation),
3. during the evaluation activity that follows the instantiation of some process characteristic.

- In general, the following categories can be identified for SPI risk factors:

1. budget and cost,
2. content and deliverables,
3. culture,
4. maintenance of SPI deliverables,
5. mission and goals,
6. organizational management,
7. organizational stability,
8. process stakeholders, schedule for SPI development,
9. SPI development environment,
10. SPI development process,
11. SPI project management,
12. SPI staff.

- Within each category, a number of generic risk factors can be identified.
- Using the risk factors and generic attributes as a guide, risk exposure is computed in the following manner:
- Exposure = (risk probability) X (estimated loss)

Critical Success Factors

- The top five CSFs are presented here:

1. Management commitment and support: SPI will succeed only if management is actively involved. Senior business managers should recognize the importance of software to their company and be active sponsors of the SPI effort. Technical managers should be heavily involved in the development of the local SPI strategy. It is not feasible without investing time, money, and effort". Management commitment and support are essential to sustain that investment.

2. Staff involvement: If SPI efforts are to succeed, improvement must be organic—sponsored by technical managers and senior technologists, and adopted by local practitioners.

3. Process integration and understanding: S/W process must be characterized in a manner that is integrated with other business processes and requirements. To accomplish this, those responsible for the SPI effort must have an intimate knowledge and understanding of other business processes.

4. A customized SPI strategy: SPI road map must be adapted to the local environment—team culture, product mix, and local strengths and weaknesses must all be considered.

5. Solid management of the SPI project: SPI is a project like any other. It involves coordination, scheduling, parallel tasks, deliverables, adaptation (when risks become realities), politics, budget control, and much more.