Chapter 1

Learning SPI in Practice

Lars Mathiassen, Peter Axel Nielsen, and Jan Pries-Heje

According to the common rhetoric of software process improvement (SPI), there are a few basic steps to improving your software development process. First, you assess your organization's current capability to develop quality software. Based on this, you derive a stepwise, focused strategy for improving this capability. You then make improvement efforts an integral part of your organization's long-term goals. The result? Both the quality of your services and the productivity of your processes increase.

SPI rhetoric is certainly powerful and appealing, and it inspires many software organizations to engage in improvement initiatives. However, even though most organizations struggle to meet expectations, many of them eventually fail.

Starting SPI is not difficult. You begin by assessing your current processes. Typically, enthusiasm at this point is high. Most of your colleagues will be eager to learn about the strengths and weaknesses in the way projects are organized and carried out. However, turning assessment insights into action is the point at which many organizations fail. Others manage to initiate focused improvement projects, only to find that implementing new ideas is very difficult (see Chapter 15). Even when you succeed in implementing an idea in an individual project, you are still a long way from institutionalizing improvements. In addition to the predictable challenges at each stage, you're likely to encounter other challenges. These include conflicts between SPI efforts and other traditional improvement initiatives, and the tendency for resources to grow scarce as the first wave of energy for SPI dies out. Clearly, SPI success involves more than simply choosing the right methods and collaborating with the best experts (see Chapter 16).

The authors of this book have all been actively engaged in SPI research and practice for several years. Our experiences have taught us what it takes to go from SPI rhetoric to success with actual initiatives. In some ways, SPI's main goal works against success: SPI seeks to change existing practices. In this change process, deeply
Chapter 1  Learning SPI in Practice

rooted values and traditions—including traditional improvement efforts—are necessarily critiqued and challenged. For most organizations, SPI is a radically different improvement philosophy, and as such it must be learned. Learning SPI can help you prepare for the changes and challenges ahead.

How does SPI differ from traditional approaches to improvement? Based on our experiences, we have identified five core SPI principles. These principles express SPI's underlying values—values that organizations must adopt to succeed with SPI. The five principles are

- Focus on problems.
- Emphasize knowledge creation.
- Encourage participation.
- Integrate leadership.
- Plan for continuous improvement.

Practicing these principles is an acquired skill. We examine each principle in more detail and discuss the factors that support and undermine them. We also provide insights drawn from the broader field of organizational learning that help explain each principle's underlying rationale.

1.1 FOCUS ON PROBLEMS

Problem solving is the essence of improvement. SPI starts with an organization’s existing practices. SPI practitioners diagnose these practices to evaluate their strengths and weaknesses; then they identify and prioritize possible improvements and establish teams to design and implement new or better processes. Practice is the SPI group’s starting point as well as their goal, and their customers are practitioners, be they software engineers or software managers. Figure 1.1 illustrates the problem orientation of SPI efforts.

This SPI principle has several immediate implications:

- There are no generally-applicable solutions. The SPI group must take into account the organization’s specific traditions, values, and capabilities.
- Many different and competing viewpoints are involved. Different actors and groups within the organization have different perceptions of the problems and of the usefulness of possible solutions.
- The ultimate measure of success is practice. Is the SPI initiative actually improving the organization’s capabilities? The SPI group must constantly ask this question to keep an improvement initiative on track.
Examples from Practice

For years, Danske Data’s methodology department had developed methods and tools to support software practices. Its first SPI assessment showed that these methods and tools were state-of-the-art. However, as Chapter 5 explains, few projects used the tools and methods. Furthermore, the methodology department did not feel obligated to ensure that its inventions were used. The SPI initiative questioned the department’s tradition of solution orientation. How well did the department members understand current practices? What did they conceive as the result of their efforts? What were their underlying success criteria? Such questioning led to many discussions and ultimately to changes in both the improvement efforts and the methodology department.

Brüel & Kjær’s project managers were skeptical about using methods and were in no way motivated to engage in improvement programs (see Chapter 6). Because project managers were key players in the organization, their attitude toward SPI was
crucial to the effort’s success. The SPI group had no choice but to build a constructive alliance with the project managers. They therefore decided to engage project managers in a dialog to identify their most immediate problems and needs.

**Key Factors**

Organizations that are learning to be problem-oriented should start with perceived problems and build diagnostic competence. To identify problems, SPI group members must understand and address the software practitioners’ perception of which practices need improvement. Two obvious strategies help here. You can analyze and formulate problems and develop improvements in direct response to practitioner perceptions, and you can engage practitioners in dialog about other, less obvious, but equally important improvement issues.

Building diagnostic competence also facilitates a problem orientation. Your SPI group should have the drive and skills to identify problems in current practices. You should develop and maintain strong relations to practice, know how to relate problems to possible causes, and relate possible improvement actions to specific problems. You can use appropriate methods—such as assessment techniques (Chapter 7) and problem diagnosis (Chapter 9)—to build diagnostic competence into your group, or you can import the competency by inviting people with relevant backgrounds and experience to participate.

Factors that undermine problem orientation include the silver bullet syndrome and a general disrespect for SPI among software practitioners. Traditional methodology departments typically believe that they can resolve problems by applying technology. Their primary strategy is thus technology push. This silver bullet approach is rarely compatible with problem orientation.

The problem-oriented approach is also undermined when SPI or the SPI practitioners lack credibility among software practitioners. A negative image of SPI among practitioners can result when the SPI group offers too little or inappropriate information, does not demonstrate useful results, or fails to interact with software practitioners.

**Broader Insights**

The underlying rationale for a problem-oriented approach to SPI conforms with general lessons from organizational learning. Argyris and Schön (1996) suggest that the real challenge in any form of organizational learning is to effectively address the gap between espoused theories and theories-in-use. Espoused theories express what people believe and think they do; theories-in-use is what they actually do. Hard as it is to admit, most of us realize on some level that self-deception, lack of discipline, and environmental factors often make it difficult to follow best practices. We keep doing what we are used to doing even though we know that other approaches are more effective.
1.2 Emphasize Knowledge Creation

As individuals and as organizations, we are constantly facing the challenge of understanding and bridging the gap between espoused theories and theories-in-use. State-of-the-art software engineering knowledge is not the only nor the most important source of learning. The key to effective organizational learning is to understand the difference between what we already know we should do and what we actually do. With problem orientation, we confront that gap. If we don’t, we risk getting stuck with general solutions and personal beliefs.


You can also use other, more general approaches to inspire your SPI initiative. One approach is Soft Systems Methodology (Checkland and Scholes 1990), which applies rich pictures, multiple perspectives, system modeling, and debates to drive complex problem-solving processes. This method takes as a starting point an unstructured situation in which problems have yet to be identified. Thus, problem owners and their different perceptions of problems play a key role in the process.

1.2 EMPHASIZE KNOWLEDGE CREATION

In essence, improvement is knowledge creation. SPI is driven by knowledge about practices and perceived needs, insights gained during the improvement process, software industry standards, and state-of-the-art methodologies and tools. SPI efforts also depend on the implicit, individual knowledge of participants. However, the general idea is to make knowledge explicit and to share knowledge.

To create useful knowledge, you must be observant and systematic. Some knowledge will, of course, remain tacit in individual skills and organizational capabilities. You should, however, make an effort to learn from practice, to make the implicit explicit, and to build widely shared knowledge about software development and SPI. Also, your SPI group should understand knowledge itself in a broad sense, ranging from experience to general, established theory. Finally, your SPI effort’s knowledge creation process should be cyclical in nature, as Figure 1.2 shows.

Your SPI group’s knowledge creation process must be deliberately designed and nurtured. We recommend that you pay particular attention to the following questions:

- How do you capture and evaluate your experiences?
- How do you combine them with other experiences and with your underlying theories?
- How is your thinking influenced by knowledge from outside the organization?
What is the quality of your knowledge?
How does your knowledge feed back into the SPI efforts?

**Examples from Practice**

As part of its SPI initiative, Danske Data established the Project Management Competence Center (see Chapter 5). The PMC Center’s purpose was to be a meeting point where project managers could exchange ideas and solutions and discuss problems, is-
1.2 Emphasize Knowledge Creation

sues, and challenges. Thus, through the PMC Center, Danske Data could better organize and use its project-management knowledge and bring in outside information to inspire project activities. However, the main motivation for establishing the PMC Center was to harness competence already present in the organization, much of which resided with a few knowledgeable and powerful project managers. PMC Center activities would help explicate this knowledge and make it available to all project managers. Danske Data also developed a project-manager training program in which experienced project managers led many of the key sessions.

Systematic was slower to realize the importance of the knowledge creation principle in its SPI effort (see Chapter 4). Several years ago, Systematic established a quality management system and was quickly certified as compliant with the ISO 9000 standards. When Systematic started its SPI effort, Systematic managers believed that the organization could reach CMM level 2 and even level 3 just as easily. It took a while for the SPI group to realize that this was not the case. Gradually, they concluded that the new change process was quite different. To succeed, common software processes had to be found, described, and institutionalized, and that would require a tremendous amount of work. This understanding emerged slowly as old knowledge was forced to give way to new experience. The company’s experience in launching the ISO certification effort required a lot of process description and some management pressure. However, the knowledge explicated in the descriptions was at that time largely shared. With the SPI effort, processes were not in place in advance and the knowledge did not even exist. Thus, new knowledge had to be introduced and shared across the organization.

Key Factors

Two factors are key to facilitating a knowledge creation approach to SPI: systematic evaluation and state-of-the-art knowledge. When you systematically evaluate software practices ("diagnose" in the IDEAL model) and specific SPI initiatives ("learn" in the IDEAL model), you create a foundation for learning from experience. This learning in turn can lead to increased understanding of what it will take to improve your organization’s software practices.

You can also import state-of-the-art theories and techniques from outside the organization. Such external knowledge can provoke your organization to change and can introduce what industry leaders consider common knowledge. The purpose is to transcend your organization’s existing software practice when internal knowledge building is insufficient (for more on this, see Chapter 12).

Knowledge creation is undermined by myths and the “not-invented-here” syndrome. Building local knowledge is important, particularly when it is done publicly. Half-baked stories of the successes and failures of various past efforts can, however,
reinforce myths and ruin good initiatives. It is therefore important that you create knowledge openly and that conclusions are tested in public. Otherwise, all knowledge will be equally important, there will be no sense of knowledge quality, and, ultimately, all knowledge will be equally meaningless and bound to remain private or localized in subcultures.

Most software practitioners are proud of their practices and results, and they often invent new approaches to deal with challenges they face. All this is positive, but if they value only in-house solutions—and thus the not-invented-here syndrome dominates—practitioners considerably reduce their ability to learn from state-of-the-art theories and techniques and thus limit improvement possibilities.

**Broader Insights**

The value of knowledge creation is clearly expressed in the SPI literature. Humphrey (1989) argues that SPI initiatives must be guided by two types of knowledge: normative models (understanding the map) and systematic assessments (understanding the landscape). Assessments are particularly important here because they provide insight into current software practices; for examples see McFeeley (1996) and Chapters 7, 8, 9, and 11.

Checkland distinguishes between the perceived world and the ideas and concepts relevant to appreciating it (Checkland and Scholes 1990). People create the perceived world through interpretations based on ideas and concepts, and their experiences of the perceived world in turn yield new ideas and concepts. According to Checkland, intellectual work is not simply a matter of making sense of the perceived world; sense-making is a purposeful activity that brings experience and theory together in a framework. Individuals conceptualize frameworks and use them to support reflection and thinking, something that leads to action.

Nonaka and Takeuchi (1995) focus on the nature of human and organizational knowledge and explain how knowledge is created in daily organizational life. Their primary distinction is between tacit knowledge and explicit knowledge. Based on this distinction, they discuss four fundamental knowledge creation processes:

- **Socialization** occurs when one person’s tacit knowledge is directly adopted by others as tacit knowledge.
- **Externalization** occurs when an organization explicates tacit knowledge as concepts and models.
- **Combination** occurs when an organization brings together different sources of explicit knowledge to create new forms of explicit knowledge.
- **Internalization** occurs when individuals adopt explicit knowledge in practice, thereby making it part of their tacit knowledge.
To create and manage knowledge successfully in your SPI efforts, you must ensure that all these processes are working at the individual and organizational level (see Chapter 14).

### 1.3 ENCOURAGE PARTICIPATION

Participation makes improvement happen. The point of SPI is to change the way practitioners develop software. However, changing human behavior is not an easy task. SPI initiatives provoke the same types of resistance to change as traditional IT implementations (Levine 1997). The factors that make IT projects successful are similar to those that help SPI succeed. The main difference between the efforts is the target customer: IT projects address the way the users work; SPI initiatives address the way software practitioners work.

One way IT projects cope with resistance to change is to encourage user participation. Early in the process, they involve the people whose behavior needs to change (see Figure 1.3). As obvious as this sounds, it is difficult to practice for many reasons. For example, if your SPI customer group is very large—thus making it impossible to involve everyone—who should participate? If you involve only representative practitioners, will they maintain the practitioner perspective over the long run, or will they themselves become change agents? Allocating time is also a problem. Software practitioners have their own work to do, and SPI work is typically logged as overhead. Given this, do you have a plan for keeping them motivated?

#### Examples from Practice

At Brüel & Kjær, participation became the cornerstone for most improvement activities (see Chapter 6). An improvement group analyzed problem reports and identified the most promising requirement specification techniques. But instead of forcing the best techniques on the projects, the support team asked project members which techniques they preferred. Table 1.1 shows the questionnaire the team used to discuss the techniques with project members. Each project then selected five or six techniques. The support team held a workshop and used some of the techniques to elicit and check a requirements specification. Not only did the result lead to better requirement specifications and better products, but also the improvement effort was so successful that many other projects asked to learn the same techniques (see Chapter 18). Letting project members pick which techniques they wanted to learn created the commitment needed for this improvement initiative to succeed.

Another example of participatory improvement comes from Ericsson Denmark (Chapter 3). Ericsson Denmark had long tried to move from one maturity level to the next but had made little progress. This changed when project managers became
Chapter 1 Learning SPI in Practice

![Diagram showing SPI effort, influence practices, and select participants.]

**Figure 1.3** SPI requires participation

---

**Table 1.1** Brüel & Kjaer's Requirement Techniques Questionnaire

<table>
<thead>
<tr>
<th>Question</th>
<th>Possible Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>To what extent will this be a change in the way you usually work?</td>
<td></td>
</tr>
<tr>
<td>To what extent do you think this will be a change in the way other people at B&amp;K usually work?</td>
<td></td>
</tr>
<tr>
<td>Please evaluate this statement: The technique described here is effective and usable.</td>
<td></td>
</tr>
<tr>
<td>How certain is it that the use of this technique will lead to the expected prevention of errors?</td>
<td></td>
</tr>
<tr>
<td>Do you have the necessary time in the project to adapt and use this technique?</td>
<td></td>
</tr>
</tbody>
</table>
involved in doing UltraLight Assessments on their own projects (see Chapter 10). Each month, every project manager assessed his or her own project and presented the results to management. Because of project managers’ commitment and involvement, Ericsson Denmark successfully advanced to the next maturity level a few months later.

**Key Factors**

As these examples show, the involvement of professional practitioners and a decentralized improvement effort are key factors in the participatory approach. The intent is not merely to persuade practitioners to practice new processes; you must involve practitioners in actually designing and developing new processes based on their own experiences and professional judgment. When the people who will use the new processes help create them, the processes are much more likely to be integrated into future practice. To succeed, you need more than just good practitioners—you need professional practitioners who are engaged both in getting the job done and advancing the profession.

Decentralizing the improvement effort facilitates participation throughout your organization and helps you to capture and account for local variations in current practices. However, inviting participation can lead efforts in unexpected directions. To see such developments as opportunities rather than threats, your organization must be decentralized because opportunities are much easier to appreciate locally. This, in turn, requires that you be able to coordinate your effort and dynamically adjust your tactics.

A participatory approach is primarily undermined by bureaucracy and firefighting. Strongly formalized assessments and centralized, management-driven SPI programs tend to make things too rigid and distant from practitioners’ daily practice on software projects. To support participation, you should limit bureaucratic arrangements and approaches. Bureaucracies are excellent ways of implementing rules and routine, but SPI follows few rules, and it focuses on problem definition and problem solving rather than routine.

Even when resources are directed toward SPI participation, practitioners are often submerged in day-to-day work, and finding time to participate in SPI activities is difficult. Also, a culture that acclaims firefighters as heroes offers individuals little incentive for investing their time in long-term improvement activities. At the start of Danske Data’s 1997 improvement effort, management directed several people to devote 30% of their time to the SPI initiative. At a first glance, this sounds satisfactory. But—not surprisingly—we quickly found that “part time is no time” (Johansen and Mathiassen 1998).
Broader Insights

The idea of participation is well established in the software profession. Mumford’s now classical work on involving users in systems development efforts (1983) has had a major impact on the profession. In Scandinavia, participation is now more or less characteristic of systems development. Bjerknes et al. (1987) and Greenbaum and Kyng (1991) provide many examples and practical approaches in support of active user involvement. Unfortunately, user participation often degrades into platitudes such as “build a prototype,” “enlist a sponsor,” and “create user-friendly interfaces” (Hirschheim, R., and Newman M., 1988). To succeed, participation must go beyond persuasion or motivation; it is a powerful strategy for building useful knowledge and must be treated as such.

1.4 INTEGRATE LEADERSHIP

Ultimately, improvements must be integrated at all levels. To succeed, your SPI efforts must be consistent with your organization’s strategy and vision of the future. Although SPI is focused on software practices, software organizations have other concerns as well. New technologies emerge, new markets develop, and alliances and mergers with other companies occur. Moreover, in many organizations, software development is not the core business but rather is one among many functions aimed at the organization’s overall goals. Given this, SPI initiatives risk getting in the way of other organizational initiatives. To prevent this, leadership should be integrated at all levels (see Figure 1.4).

Our primary concern here is management’s ability to use leadership to motivate and set direction. When your SPI vision and your organization’s overall strategic vision are aligned, management can be integrated at all levels. In our experience, a good vision is closely linked to how leaders perceive the existence of people in the organization. It is not enough to build a vision. As a leader, you must base your vision on the organization’s reality and present the vision in a way that motivates others to both understand and believe in it.

Examples from Practice

At Danske Data, top management endorsed and supported the SPI initiative from the start. For example, the CEO spoke at the workshop where Danske Data’s standard for project management was created (see Chapter 5). This ensured that the initiatives around project management—such as establishing the PMC Center—were consistent with the overall company strategy. However, it is also fair to say that in this case, middle management did not share the SPI vision and generally ignored the SPI effort’s existence.
Another example of leadership comes from Systematic. From the beginning, Systematic's CEO was keen to achieve CMM maturity level 3 (see Chapter 4) and to include customers in the improvement process (see Chapter 13). These aims were woven into all the organization's plans and actions. Because the organizational vision and the SPI vision were aligned, Systematic could set ambitious goals and move forward. Furthermore, Systematic's CEO walked his talk. When an assessment identified project management as a problem, the CEO invited several suppliers to compete to deliver a project-management training program. Within a year, more than half of Systematic's project managers had completed 12 days of intensive training.
Chapter 1  Learning SPI in Practice

**Key Factors**

The most important leadership quality is the ability to communicate. Managing, controlling, and monitoring do not a leader make. Leaders need vision and the skills to communicate that vision throughout organization. According to Kotter, any type of change requires 75% leadership and only 25% management (Kotter 1996). SPI visions, plans, and achievements should be widely communicated. To maintain attention and commitment, you should produce results regularly and disseminate them widely.

Leadership is undermined by a lack of management commitment and the Balkan syndrome. At Ericsson Denmark, management was committed, but the commitment was wildly unfocused. At one point, there were six key process activities (KPAs) on the SPI agenda, plus several critical success factors and a few “vital few actions” (see Chapter 3). This scattered commitment made the SPI effort confusing at both the project and the organization level. It was not until the responsible manager outlined a simple and clear strategy—achieve maturity level 2 before summer—that change and improvement began. Such a lack of management commitment is not limited to top management; it might also apply to middle management or any type of supporting staff.

The Balkan syndrome is common in software organizations. It occurs when each group or department has its own way of doing things and develops individual professional standards. Without strong leadership, improvement efforts tend to diverge and common commitment disappears. Although you must take variations and differences into account, you should do so only when necessary. When projects learn from each other and processes are reused across the organization, you can better focus energies when difficult challenges arise.

**Broader Insights**

This principle is grounded in strategic theories that consider not only an organization and its environment (see Ansoff 1988) but also internal factors such as organizational structures, production processes, and technology (c.f. Chandler 1962, Scott 1987). Furthermore, we view the strategic plan not as the most important part of strategy but rather as an outcome of an ongoing process of integration and reorientation. For SPI to succeed, it must be an integral part of the organization’s strategic leadership.

Organizational leadership involves the ability to build a shared vision and to identify prevailing mental models that need to be challenged. In a learning organization, leaders “are part of changing the way business operates, not from a vague philanthropic urge, but from a conviction that their efforts will produce more productive organizations, capable of achieving higher levels” (Senge 1990).
1.5 PLAN FOR CONTINUOUS IMPROVEMENT

Improvement should be a continuous effort. It is easy to imagine the end of an SPI project, when you count your successes and failures. It’s equally obvious that improvements are in some sense endless: As you alleviate some problems, others become visible. Both views make sense.

Organizing your SPI efforts as projects that have an end point lets you stop, step back, and evaluate your SPI initiative and its progress. When an SPI project ends, you can move responsibility for the new processes into the organization itself. Nonetheless, as Figure 1.5 shows, SPI initiatives are necessarily ongoing because there are always new problems and challenges, and solutions to old problems must be maintained and further developed.

Examples from Practice

Although our experiences span only a few years, it is clear to us that a continuous approach is important. At Danske Data (Chapter 5), recent discussions have focused on

![Figure 1.5 SPI should be continuous](image-url)
how to organize the SPI effort in the coming years. Danske Data has long had a large methodology department. Initially, the SPI group attempted to dissociate itself from this department because it was not well regarded by software practitioners. However, the SPI effort has been gradually reorganized to include the methodology department so that it can fill the growing demand for maintaining new software processes. In some ways, the methodology department and the SPI initiative have merged. In other respects they are distinct initiatives. In any case, the SPI effort has now become a stable part of the organizational structure.

Another event at Danske Data supports this view. After two and a half years of SPI effort, a Bootstrap assessment showed that the company had reached its goal of maturity level 2. The CEO then decided to go after level 3. Evidently, the SPI effort will be ongoing for some time. A question remains, however, as to how far Danske Data will or should go on the conventional maturity ladder. Metaphorically speaking, it is not important to reach the summit. What matters is that you stay on an ascending path that is appropriate for your organization. Reaching one goal thus creates another, higher goal that has meaning for your processes and their context.

**Key Factors**

The key factors in a continuous approach are stepwise improvement, top-management commitment, and a sustainable improvement organization. Maturity models such as CMM, Bootstrap, and SPICE (Software Process Improvement and Capability Determination) embody the essence of stepwise improvement. All experiences suggest that such ladders take years to climb.

A continuous approach thus requires commitment from top management. A local or bottom-up initiative can be effective for a while, but it easily loses momentum. As time goes on, the focus on SPI initiatives will decrease unless they succeed in becoming part of the organization’s strategic thrust.

To successfully climb the maturity ladder, you must install and maintain impeccable improvement processes. To do this, you need a sustainable improvement organization that is adaptable to changing circumstances. Management must directly allocate resources to such an organizational unit and must help it create and maintain organizationwide SPI awareness.

Factors that undermine the continuous approach are inadequate results and a marginal SPI effort. If results are not documented or the return on investment is poor, your SPI effort is unlikely to survive past the first project. In most cases, you cannot simply document improved maturity on a normative scale. Practical, convincing results are needed, and it is typically best to establish a simple metrics program that can help you argue for continuing your efforts (see Chapter 17).
Even when an SPI effort is showing results, organizational politics or other organizational issues can still marginalize it. Brüel & Kjær’s SPI effort went from marginal to central and widely known and back to marginal (see Chapter 6). At this point, it looks as if Brüel & Kjær will most likely discontinue its six-year SPI effort.

**Broader Insights**

The IDEAL model clearly expresses SPI’s continuous nature (McFeeley 1996). The main theory underlying the continuous approach is Humphrey’s CMM (Humphrey 1989). The CMM is specific to software development, but it shares a basic structure with Crosby’s five-level model of quality systems (Crosby 1979). The idea of continuous, stepwise improvement is also common to many other quality models.

In his work on Soft Systems Methodology, Checkland has been particularly clear on how alleviating some problems makes way for others (Checkland and Scholes 1990). Effective problem-solving approaches are therefore continuous. Checkland and Scholes also distinguish between two types of problem solving: intervention and interaction. The interventionist mode is external and uses problem solving to structure an internal inquiry. The interaction mode is internal and uses problem solving to make sense of experience. The latter mode is similar to the continuous approach, whereas the interventionist mode corresponds more with a focused collaboration with SPI consultants over a limited time.

Theories of organizational learning view learning as continuously present and as a significant factor in organized work and behavior (see, for example, Argyris and Schön 1996, Senge 1990).

**1.6 LEARN TO IMPROVE**

Our five principles are strongly interdependent. If your organization fails to practice one of them, you seriously reduce your chances of successfully implementing the others. For example, if you fail to focus on problems, the following occur:

- You cannot create new knowledge based on your organization’s software practices.
- A primary motivator for having practitioners participate disappears.
- Management is more likely to reduce improvement initiatives to traditional support functions with little strategic importance.
- You are left with only an overall concern for improving software practices, and that concern cannot drive a continuous, long-term improvement effort.
If you fail to adopt other principles, similar weaknesses will result. The five principles are a coherent philosophy of SPI that we have developed through practice. The underlying values are different from or even contradictory to the values of conventional improvement approaches. It is therefore not simply a question of deciding to practice SPI. Learning SPI is a demanding process; it questions personal and professional beliefs and challenges existing traditions.

If you start your SPI initiative as a series of projects, you can maintain your focus and allow for dynamic changes as your particular process evolves. Above all, SPI requires committed, patient participants. To learn SPI, you must be prepared to improve both software practices and existing improvement traditions. To prepare yourself for such a challenge, in Chapter 2 we offer an overview of the existing SPI literature and recommend that you take advantage of the knowledge it contains. Ultimately, however, each organization must find its own path to SPI success. Your primary and most important role will be that of change agent. The best change agents avoid becoming dogmatic SPI followers. Rather, they learn what they can, listen to the many voices within their own organization, and, based on both, they chart the best path toward change.

1.7 REFERENCES


