ELSEVIER

The Journal of Systems and Software 47 (1999) 139–148

The Journal of Systems and Software

www.elsevier.com/locate/jss

Towards an evaluative framework for software process improvement models

Hossein Saiedian [a,*], Kalyani Chennupati [b]

[a] Department of Computer Science, University of Nebraska at Omaha, Omaha, NE 68182-0500, USA
[b] Metamor Information Technology Services, 4000 McEwan Road South, Suite 200, Dallas, TX 75244, USA

Abstract

Today the modus operandi for software development is heavily process-oriented. This is based on the premise that there has to be a quality process in order to produce quality software. There are number of quality models for software development called Software Process Improvement (SPI) models, which address this important issue. As these models differ in their characteristics it is important that there be a basis to evaluate them effectively. Such an evaluation is important from the perspective of understanding the particular model in terms of its structure, its benefits, etc. For the evaluation to be comprehensive and systematic, it is important to have a framework (or benchmark). This article defines such a framework which includes the most important criteria for the evaluation of the SPI models. © 1999 Elsevier Science Inc. All rights reserved.

1. Background

In order to improve a software process, a *software process improvement* (SPI) (Zaharan, 1998) model is needed. The objective of a software process model is to provide a framework for producing software products according to the plan while simultaneously improving the developer's capability to produce better products (Humphrey et al., 1989). A software process model can be used by an organization to assess its *maturity* and to identify and prioritize the highest areas for improvement. The term maturity in this context refers to the capability of an organization in attaining a stable, defined, repeatable and optimized level of software development. Results of an assessment form the basis for an action plan for organizational self-improvement.

There are number of SPI models. Examples include the Capability Maturity Model (Paulk et al., 1995) the ISO 9000-3 series of standards for software (Kehoe and Jarvis, 1996) the AMI model (Pulford et al., 1995) the Bootstrap model (Kuvaja et al., 1994) the SPICE model (El-Emam et al., 1998) and few smaller ones. When an organization commits itself to process improvement, it looks for an SPI model which will be suitable in terms of its business needs. The choice of a particular model depends on the type of organization, its business needs and its business goals. If an organization is small, it may not be able to invest a great amount of money and resources on an expensive process improvement program. On the other hand, large companies can easily afford the overhead costs and launch an extensive SPI program. Hence, it is imperative that an organization chooses an SPI model which closely represents its aims and goals and effectively uses it to launch its process improvement efforts.

In order to evaluate existing SPI models most appropriately, a framework is needed. Such a framework will help an organization to choose the most suitable model for its SPI program. The objective of this research is to develop, define and justify the components of such a framework. The framework will answer the questions such as what model to use, when to use it, what the potential costs will be and what resources and management commitments are necessary to successfully implement it.

A software organization's motivation to improve its software process builds from a business need such as strong competition, external regulation or a call for increased profitability (Paulish and Carleton, 1994). Software assessments help an organization move towards its goal of software process improvement. Although assessments of any kind are only a beginning (Bush, 1991; Koltun, 1992) they form the crux of the SPI program of an organization. Assessments as such are discussed in more detail in the next section. After

[*] Corresponding author. Tel.: +402-554-2849; fax: +402-554-2975; e-mail: hossein@cs.unomaha.edu

0164-1212/99/$ – see front matter © 1999 Elsevier Science Inc. All rights reserved.
PII: S0164-1212(99)00034-5

assessing its current practices and process maturity, an organization often initiates approaches to improve a software development process. A common process improvement approach given by Paulish and Carleton (1994) is as shown in Fig. 1.

Software improvements have to be carefully planned and executed systematically. Improvement programs cannot be random in order to be successful. Most successful companies use a pattern for process improvements which consists of a six stage improvement program after the initial activity of an assessment and a baseline (Jones, 1996). The sequence for a process improvement program can be generalized as shown in Fig. 2.

A formal process assessment and a quantitative baseline of current productivity and quality levels marks the beginning of all SPI strategies. Assessments serve as a diagnosis of applying metrics to measure impact evaluation of business needs motivation to improve implementation of improvement model selection of an improvement model assessment the organization's strengths and weaknesses and the baseline provides a basis for productivity, schedules, cost, quality and user satisfaction. The first improvement stage involves the training of the management. This stage involves equipping the managers with the knowledge of the critical technologies which are required for SPI. The second stage of processes and methodologies concentrates on approaches for dealing with requirements, design, development and quality control. Implementation of new tools and technologies fits into the next stage of SPI. At this time when the processes and methodologies have been targeted, the organization needs to acquire improved tools and explore new technologies. The infrastructure and specialization phase involves the establishment of special teams or groups for handling three testing, maintenance, integration and configuration control. The next stage of SPI involves focusing on reusability in order to bring down the costs. An organization which reaches the sixth stage of SPI stands out as an effective organization which has successfully implemented its SPI program.

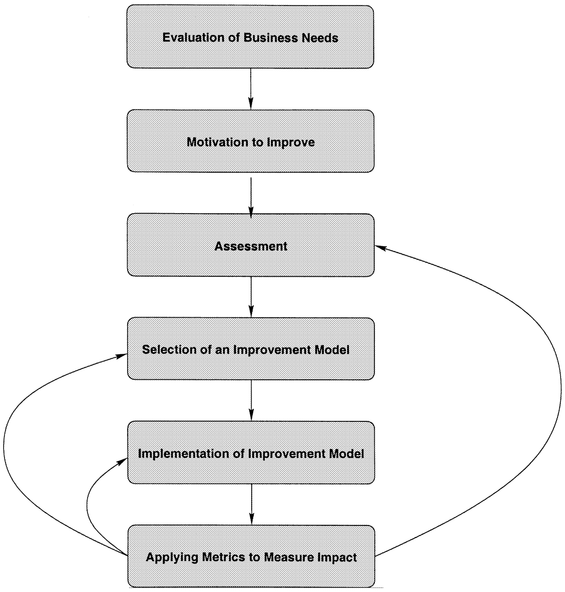
Evaluation of Business Needs
Motivation to Improve
Assessment
Selection of an Improvement Model
Implementation of Improvement Model
Applying Metrics to Measure Impact

Fig. 1. Process improvement approach.

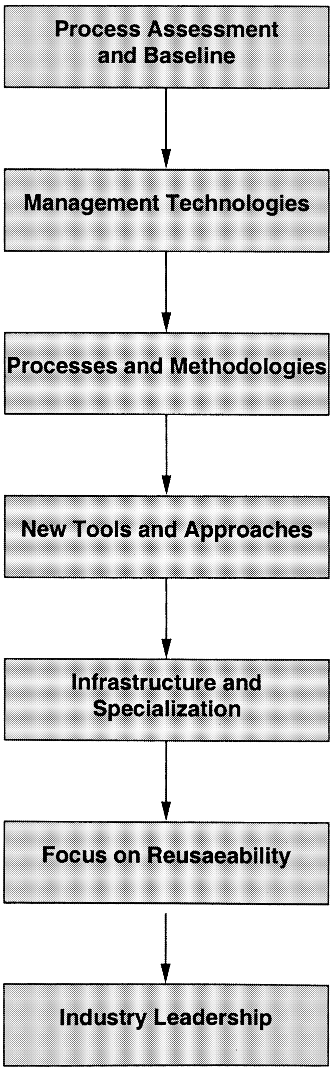
Process Assessment and Baseline
Management Technologies
Processes and Methodologies
New Tools and Approaches
Infrastructure and Specialization
Focus on Reusaeability
Industry Leadership

Fig. 2. Optimal sequence for software process improvement (Jones, 1996).

Dion (1992) stated that the time was right for organizations interested in improving their software development process to plunge into it. Many organizations like Raytheon have a process improvement program in place. It is now recognized that traditional engineering management methods work for software as they do for other technical fields (Humphrey, 1992). In view of this, process assessment helps software organizations improve themselves by identifying their critical problems and establishing improvement priorities. A software process is a feedback process where things happen are observed and the observed information is used to control the process or to make changes to the product. According to Cheek (1995) a software process behaves like a feedback system at all levels in terms of the actual process of development and in terms of user reaction.

An assessment acts as a diagnostic tool to aid organizational improvements. In a software organization, the assessment starts with the senior manager's commitment to support software process improvement. The objectives of an assessment are as follows.

- To provide a clear and factual understanding of the organization's state of software practice.
- To identify its major problems, i.e. to identify the key areas of improvement.
- To initiate actions to make these improvements.

A software process assessment acts as a review of a software organization which can be used to advise its management and professionals on how to improve their operation (Kitson and Humphrey, 1989). The assessments are conducted by a group of software professionals who have assessment experience or training. The main aim of assessments is to provide guidance to local managers and professionals who are already committed to improve their own operation. The guidance which is provided can be in terms of what to do and how to do it. Assessments enable the identification of highest-priority areas for improvement and to provide guidance on how to make these improvements.

During an assessment, an organization's process is reviewed in comparison to some vision of how such processes should be performed. As in many technical activities, a sound assessment requires that the basic requirements are met. A good assessment involves a competent team, sound leadership and a cooperative organization. In addition to these, there are some special considerations which should be viewed and given importance to. These are (Humphrey, 1989):

- The need for a process model as a basis for the improvement and assessment.
- Requirement of confidentiality.
- Senior management involvement.
- An attitude of respect for the views of the people in the organization being assessed.
- An action orientation.

Although all of the above are important, we will concentrate on the first item, i.e., the need for a process model as a basis for process improvement and assessment. An assessment requires a strong foundation on which it can be built. In view of this, there is a need for a process model which can act as a basis for the assessment. This point being the crux of the assessment process, is further discussed in the next section. The purpose of an assessment is to support the organization's improvement program and not to report its problems to higher management. If this is not so, it becomes tough on the part of the assessment team to conduct assessments that uncover the real issues. Confidentiality is required as it permits assessors to talk to people at all levels of the organization. Confidentiality at all organizational levels ensures that no single project or individual is identified with any specific problem.

When an organization plans to improve its software development process, it looks for an SPI model which will be suitable in terms of its business needs. It may also happen that the presence and success of an SPI model influences an SPI initiative in an organization. For example, the use of Capability Maturity Model (CMM) (Paulk et al., 1995) by the DoD contractors in the US has influenced its spread to many commercial organizations. Although it is true that these organizations see the model as providing an opportunity to bid for the DoD contracts, many organizations have used the model with the ultimate aim of improving their process and have been successful in their attempts. The choice of a particular model depends on the type of organization, its business needs and its business goals. If an organization is small, it may not be able to invest a lot of money and other resources on the improvement programs. On the other hand, large companies can easily use the overhead money and launch an extensive SPI program. Hence, it is imperative that an organization choose an SPI model which closely represents its aims and goals and use it to launch their SPI efforts.

Software process improvement models share a common concern regarding software quality and process management. Since it is not clear which of these models is most effective in achieving their shared objectives, it is valuable and timely to provide an objective evaluation of these models and, to compare and contrast their features for quality software development. Because there are many legitimate areas for comparison, an "evaluative framework" is needed that highlights the most visible elements for evaluation purposes.

A framework that evaluates different SPI models can be used in an effective way by an organization. Such a framework will help an organization to choose a specific model or models for its SPI program. The basic questions that an organization would look forward to be answered when it wants to choose a specific model could fall under a number of categories. The framework that has been created as part of this work attempts to answer these categories of questions. Table 1 shows the framework that has been created. Questions fall under four different categories: *WHAT*, *WHICH*, *HOW* and *WHERE*.

WHAT Questions. The *WHAT* category of questions deals with what the model is in terms of its goals, the base structure, the role of the management, the kind of metrics used and the benefits of the model. There are total of five questions under this category: What are the goals of the model? What is the underlying structure? What is the management role? What are the benefits of the model?

Table 1
The components of a framework for evaluating SPI models

Categories	Category elements	Description
WHAT	Goals of the model	The objectives of the model
	Underlying structure	Base structure of the model
	Management role	The extent of management involvement
	Use of metrics	The kinds of metrics used
	Benefits of model	The benefits gained through the use of the model
WHICH	Underlying models used	Which other models, approaches and standards have been used to develop the model and the interlinks between the different SPI models
HOW	Rating process	How the model rates the capability of an organization
	Organizational impact	The impact of the model on an organization
WHERE	Scope and domain	Which type of software organizations use the model and in which countries it has been used

WHICH Questions. The *WHICH* category of questions dealing with which other models, approaches, principles, and/or standards have shaped the development of the SPI model.

HOW Questions. The *HOW* category of questions attempts to answer how the capability of an organization is rated, how the capability is found out and how the model can be used for an effective SPI program. Questions under this category include: How the model rates the capability of an organization? and How the organization can use the model for software process improvement efforts?

WHERE Questions. Finally, the *WHERE* category of questions attempts to answer the question of which types of organizations have used the model and in which countries it has been used.

The above categories of questions are reasonably exhaustive when they are applied to an SPI model. The framework attempts to address the issues that are most important to be considered in order to enable an organization to choose an SPI model or models. The objective of this article is to describe and justify individual elements within each category of questions and to discuss why a particular question is important for consideration in the framework. In a concurrent study, we are using this framework to evaluate six different software process improvement models and standards. In the rest of this article, we define and justify each element of the framework and elaborate on its significance.

2. An evaluation framework

In this section, we introduce the elements of the framework and briefly describe and justify each one.

2.1. Goals of the models

Each SPI model has some objectives and satisfies some goals. The development of an SPI model is influenced by the requirements of the software industry. The software industry has come a long way from being worth a few million dollars to trillions of dollars. Software industries build software ranging from a few KLOC to hundreds and thousands of KLOC. The main reason that software costs so much is not only due to the cost of the software itself but, due to the fact that software producing organizations have more often than not seen schedule overruns. A recent study of 13 DoD contracts showed that 24 months project overshot its schedule commitment by 20 months (Humphrey, 1989). Another example of a software crisis is more closer to home. The fiasco of the Denver International Airports Baggage Handling System is seen as a very good example of software complexity.

Technology is one aspect of developing software. But, unless an organization has a stable process to develop a software product, it will not be able to reap benefits of the introduction of new technology into the development process. Hence, an organization has to concentrate on its process and try to improve it. The focus on software process and software process improvement has led to the development of many SPI models, approaches and methodologies. In this work we consider all of them as models for software process improvement. Even though the main focus is improvement of the software process, each SPI model has unique objectives and goals. For example, a model like the CMM (Paulk et al., 1995) was developed to serve the needs of the DoD, Bootstrap (Lebsanft, 1996) was developed with European software organizations in mind which have very special features, and the AMI (Pulford et al., 1995) model was developed to totally concentrate on providing metrication to a process improvement program. Each of the SPI models has been developed with its goal(s) in mind and hence, there are variations in their base architecture. Therefore, in order to understand the base architecture of the SPI models, it is necessary that an organization has a clear and unambiguous picture of the goal(s) and objectives of each model.

2.2. Underlying structure

The base structure of an SPI model helps an organization to view its SPI program as a process by itself. An organization must have a clear picture of how a particular SPI model is structured and what its key components are. In order to use an SPI model, an organization should understand each and every component of the base structure of the model and how it can be applied to their organization in particular. Each SPI model may attack the problem of SPI in a different manner. For example, the structure of the CMM is based on maturity levels which are made up of Key Process Areas (KPAs) and Key Practices (KPs). It is important that in order to be at a maturity level, an organization has to satisfy the goals of each KPA at that maturity level. The CMM identifies certain KPs that need to be done to be at a particular maturity level. Hence, an organization has to satisfy these KPAs. It may choose to do other activities that are not specifically mentioned in the CMM KPAs, but it is required to satisfy the given KPAs at a particular maturity level.

SPICE (El-Emam et al., 1998) on the other hand has processes which are not concentrated at one level but, may span many levels. So, if an organization wants to use the SPICE model for SPI, it has to understand the basic organization of the models and use it accordingly. Trillium is another SPI model in which the Capability Areas span several maturity levels. The AMI SPI model has a base structure very different from SPICE or CMM. This is because the approach aims at helping an organization create a measurement program to guide SPI efforts and to keep track of the SPI activities.

The terminology used in the base structure of each of the SPI models used here is also different. If an organization wants to make use of one or more SPI models for its SPI program, it needs to make sure that the models do not follow wholly two different approaches. Such a choice would create confusion and the purpose of the use of the models would be defeated.

2.3. Management role

Why is it important that the management be involved with process improvement? If an organization can get the best technical people, it should be able to produce quality software. In addition, an organization can introduce new technology which looks promising for their development process in order to produce quality software. But it has been seen that the situation is far from true. Even with the best people and the best technology, an organization is not guaranteed that the product that it produces will be of the required quality. According to (Dewey, 1988) the most important software productivity and quality improvements are management and not technology driven. A recent study by Goldenson and Herbsleb (1995) showed that organizations in which the managers actively monitored the progress of process improvement showed marked success in their SPI efforts. Such management commitment was seen to be considerably less common in the organizations with less successful improvement programs. Senior management oversight plays a very important role in process improvement.

The management is responsible for providing direction to the SPI effort. The ultimate need for improving the process comes from the fact that the organization always strives to increase its profits and tries to maintain or increase its market share. The initiative for the process improvement comes from the management. Hence, it is important that the management guides the improvement effort and provides the necessary funding and resources for an effective and successful implementation of an SPI program. The management needs to clearly state the process improvement goals and ensure that they are well understood throughout the organization.

An organization that wishes to start an SPI program by using an SPI model should know where it should concentrate on, what its responsibilities are and how well it can give directions to the improvement efforts. Each SPI model may have a different concept of the role of the management. Some may require specific responsibilities on the part of the management. For example, ISO 9000-3 (Kehoe and Jarvis, 1996) requires that the producer management as well as the customer management be involved in a Quality System. AMI which totally concentrates on measurement activities for SPI, requires the management to initiate the measurement program and take part in the formation of the goal tree. CMM requires that the management be actively involved at all levels of the maturity scale and this is addressed through the KPAs at different levels.

With an SPI effort, an organization can look forward to improving its process and the quality of its products. The management of the organization should take steps to reward its employees so that the profits accrued through SPI percolate to them. This helps to sustain the improvement efforts and ensures that process improvement is continuously striven for. In addition to providing resources and funds, the management should help to prioritize the improvement activities to ensure that improvements are not done in a haphazard manner. The role of the management is hence, very important and the management of every organization which wants to improve its development process should strictly follow the concepts of the SPI model.

2.4. Use of metrics

Measurement is a critical issue in process improvement. If an organization wants to control its process, then it has to measure it. DeMarco (1982) emphasizes the need for measurement by stating that "You cannot control what you cannot measure."

Many organizations do not have the necessary infrastructure in place to control their processes. And this is of a greater concern if the process needs to be improved. An organization needs to know what measurement is and why it is so important for process improvement. Fenton (1991) gives a clear and formal definition of measurement:

> Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

With respect to software development, the entities are the different phases of the development process and the attributes of these phases include cost, time, etc. The quality attributes may not be congruous with each other. For example, inspections during the requirement phase will lower the rate at which defects are found during the testing phase, but it will increase the cost of the requirement phase. It is hence, important that different weighing factors are used for different quality attributes keeping in view the type of the software and the customer. For example, for large customers with real-time processing environments, performance and reliability may be the most important attributes, while for customers with stand-alone systems and simple operations, ease of use, installation and documentation may be more important.

A measurement program does not stop once the measurement areas/attributes are identified. Rather, the identification phase is just the beginning. An organization has to identify the areas of measurement, do the actual measurement and use the data to control the process. In order to successfully control the process, the identified areas of measurement should be authentic and useful. It would be disastrous if an organization starts to measure anything and everything with the hope that it will be useful at a later time. The measurement objectives should be clear and well-defined. Once the data is collected, it has to be transferred into pieces of information which can be used effectively. Therefore, analysis plays an important role in a measurement program. A measurement program should be analysis-driven rather than data-driven.

Measurement has to start from requirements stage onwards. Requirements pose a major problem to software development. According to Jones (1992) 5% or more of all software defects are requirement errors. Hence, a software producing organization which does not address the quality of the requirements is more prone to produce poor-quality products. There are several other legitimate areas for measurement, like cost of the project, whether the schedule commitments are being met, customer satisfaction, etc. Customer satisfaction is emphasized in SPI models like Trillium and ISO 9000. ISO 9000 is totally customer-oriented in that conformance would tell the customer that the organization is doing what it says it is doing. Measurement of customer satisfaction is not an important area of measurement in CMM. Hence, the areas of measurement are different for the SPI models with some areas of overlap for each. Due to this it is important to know to what extent measurement is important and which areas are addressed to be measured in each SPI model before it can decide on the suitability of a particular model. With a valid measurement program, an organization can know where it is and where it wants to go. The data gathered can be used in a scientific way to help in process improvement.

2.5. Benefits of model

A software organization would be more willing to take up an SPI effort using a specific SPI model if organizations which have already used the model have shown some economic benefits. In order to maintain or improve their market share, an organization will surely look for a return on its investment. The economic benefits from SPI efforts that a management of any software organization looks for is a return on their investment (ROI). Every model dealt with in this work has shown evidence of ROI, though not typically in the form of dollars returned for a dollar invested. Although some organizations publish ROI data in terms of quantitative values, there is a general dearth of published data in the public domain.

There are many reasons for the unavailability of quantifiable economic benefits of SPI efforts. An organization may not have the necessary infrastructure to measure the benefits in terms of dollars returned for a dollar invested. Some organizations may be more interested in economic benefits of increased productivity or reduced rework. There are examples where organizations have been able to show improvements in terms of ROI (Dion, 1992) reduced defect rates (Butler, 1995) and increased productivity (Butler, 1995). A recent study by Brodman and Johnson (1996) shows that the US industry's focus is on effort as the primary input on the investment side and on meeting cost predictions, improving cost performance and staying within or under budget on the return side.

The management of any software organization has the responsibility to achieve a return on its investment. Any management would be reluctant to act if credible estimates of what the return would be is not known. Many organizations like Raytheon (Dion, 1992) have shown encouraging return on investment values. Not all the models discussed in this work have become full-edged models for SPI. For example, SPICE is still under a standardization phase and is expected to become a full-edged SPI model in 1998. Besides, only those organizations which have been successful in implementing an SPI program would be inclined to release data on their ROI than organizations who have not been successful or have made only small improvements. For example, there are a number of successful cases reported by organizations which have used CMM for their SPI and only a few among these have published or released their ROI findings (Herbsleb et al., 1994). Nonetheless, these success cases show that a rigorous approach to SPI using the model that was used does provide tangible benefits in terms of ROI. Such cases should serve as an encouraging factor for the use of the models for SPI. The benefits of an SPI model can also be calculated in terms of cost avoidance such as rework, duplication of function etc., increased productivity, reduced schedule time or improved quality.

2.6. Underlying models

Each of the SPI models has been developed to cater to a particular type or types of organizations. Quality was and is still perceived as one of the goals of a software producing organization. It is only when an organization can produce quality product in time that it can maintain its competitive edge and aim at getting higher profits. A model like the CMM was developed on the basis of quality principles of Juran, Deming, Shewhart and Crosby. Hence, these principles have to a large extent shaped the design of the CMM base structure. A number of other models like Bootstrap, SPICE (Coletta, 1995) and Trillium (Coallier, 1995) have used the CMM for their development and hence, one can see the flavor of CMM in these models. When an SPI model is developed, the developers see what needs to be added or included in order to make sure that the SPI model serves its purpose. It is always a good idea to use a previously developed SPI model or use quality principles to build a new model. Of course, the base SPI model should be applicable to the problem at hand and should be complemented with additional ideas/principles in order to make the new SPI model applicable and useful. Otherwise, the development of an SPI model would involve reinventing the wheel. The development of the new SPI model would involve investment not only in terms of money, but also in terms of effort, time, etc., and at the end it might not serve any unique role.

The models in this work have used existing models for SPI during their development. It is important to have a good perspective of why a particular model, approach, principle, or standard was used to develop a particular SPI model. Not only does it give a historic perspective to the model but aslo helps us to understand why a particular concept has been included in the model and what purpose it is supposed to serve. This also helps an organization to decide whether a particular SPI model can be used to implement an SPI program or whether the model needs to be tailored in order to fit to its requirements.

2.7. Rating process

If an organization wants to improve its software development process, it needs to know where it stands. A clear vision of its goals is important, but the baseline is also equally important. An organization may strive to improve the quality of its products by trying to improve its process, but, if it does not know where it is presently standing then it may not be able to plan its improvement activities. Process improvement has to be done in small evolutionary steps which progress in such a way that the organization has a clear picture of the process and it can see how the process is being improved. This is important as the continuous use of an SPI model can be justified only when it improves the process and this improvement is visible to everyone who is involved in the initiative.

The scale which is used to rate the capability of an organization is hence, very important. This scale can be in the form of a staged structure as in the CMM or it may be in the form of a coherent whole with activities to be done distributed in different steps as in Trillium. The rating scale which is used in an SPI model can be used to rate the maturity and can also be used to provide a vision of what the process should be. For example, in the case of the CMM, the rating scale which is used rates an organization as being either at the Initial, Repeatable, Defined, Managed or Optimizing level. An organization which is at the lowest level of the CMM scale (Initial level) has an ad hoc process in place while an organization at the highest level of the CMM scale (Optimizing Level) has fully optimized its process (and continues its optimizing status) and can be considered as a industry leader. This staged structure of the CMM shows that an organization has to fulfill all the requirements of a particular level in order to be at that level. The same concept is used in the Bootstrap model with the exception that each level is further divided into quartiles. This is beneficial as an organization may wish to concentrate on reaching sub-levels before climbing to the next level on the maturity scale.

Trillium and SPICE use a similar concept except that the key activities are distributed over a number of levels. For example, a KPA in the case of CMM is nested within a particular maturity level whereas a Capability Area in Trillium may span different Trillium levels. AMI on the other hand is an SPI model which does not rate the capability of an organization. It does use a rating process during the “assess” phase, but this process could be the CMM, Trillium or any other approach.

An organization needs to know what the rating scale is for a particular SPI model which it is using or plans to use. This rating scale can be used to guide the organization in its improvement program and an organization which is at a lower level on the rating scale can try to reach higher levels and ultimately aim at being industry leaders.

2.8. Organizational impact

Each of the SPI models may have a different method by which the capability of a software organization can be measured. As the ultimate aim of an SPI model is to guide an organization to improve its process and improve its quality, an SPI model has to provide a meaningful means by which the current capability of an organization can be found out. The use of the model should also enable the organization to set its targets and enable it to reach a particular capability. After an organization finds out where it stands with respect to its process, it can aim to improve the process. An organization should know how the capability is actually measured as how the subsequent SPI efforts can be guided through the use of an SPI model.

Capability determination is the starting point for any SPI initiative. An SPI model can be used to assess the current process. For example, CMM provides two methods by which capability of an organization can be determined. The SPA method and the SCE method can be both used towards the end of the software capability determination. Even though the main purpose is to determine the capability, the two methods differ in how the assessment is carried out and to what extent the organization has to be involved in it. On similar grounds, Trillium also provided methods which can be used by an organization for its internal process improvement efforts or can be used for determining the capability of a contractor organization for award of a specific contract. On the other hand, AMI does not provide a capability determination method. Rather, it uses an existing SPI model which supports capability assessments. Hence, if an organization wishes to use the AMI model it has to decide on which SPI model to use in the "assess" stage of the model. Although the CMM SCE can be used, a CMM SPA can be used effectively during the "assess" stage of AMI.

As the different SPI models may have different methods by which capability is determined, an organization should understand how the SPI model works and how it can be used for process improvement efforts once the capability is determined.

Once an organization commits itself to improve its software, it will have to initiate a process improvement program. Selection of an SPI model, which is based on how well it suits the business goals of the organization, to what extent it is cost-effective, and to which types of organizations it can be applied to, is crucial for the success of the improvement program. As these models differ in their characteristics it is important that there be a basis to evaluate them effectively. Such an evaluation is important from the perspective of understanding the particular model in terms of its structure, its benefits, etc. This article defines a framework which addresses the most important criteria for the evaluation of the SPI models. The framework provides a basis for the evaluation and selection of the SPI models.

2.9. Scope and domain

The Trillium model is the best example of an SPI model which was developed with a particular type of organization in mind. Specifically, a telecommunications software producing organization can use the model for assessing its contractors and for its own SPI programs. Hence, this model can be used by a specific type of organization. This is seen by the fact that the Trillium model is known to be used as an SPI model by Bell Canada and Nortel, which are two large telecommunications-oriented software developers.

The scope of an SPI model is, hence, very important. If a model has been used specifically by a particular type of industry, then it is possible that the model will be used by similar kinds of organizations. It could also happen that other kinds of organizations may successfully use the model. CMM is an example of an SPI model which was developed for a particular kind of organization (i.e., U.S. Department of Defense) but, is also used by other commercial organizations both in the US and outside.

The domain of an SPI model also influences its success. If an SPI model is used in a number of countries or even worldwide, then organizations would not be hesitant in using the model for their SPI programs provided positive benefits were seen through the use of the model. For example, the benefits accrued through the use of CMM was seen as its positive point. The model's usage area has spread to many European countries and even to countries in Asia. The acceptance of CMM in countries other than the US has given a positive thrust to its popularity. On the other hand, Bootstrap has not been so successful. Although Bootstrap was developed for the European software producers, many US and Asian software companies have used the method. But, Bootstrap did not pick up in these countries. This can be seen in the context of companies which used the model in India. The German government had sponsored the assessments to be done by the Bootstrap model, but, after that the companies showed limited interest in the model. A similar situation is seen in the US where the companies which have used the model wish to remain anonymous.

The scope/domain of an SPI model gives a good picture about its usage in terms of what kind of industries have used it and to which countries it has spread its reach.

3. Conclusions

When an organization commits itself to process improvement, it looks for an SPI model which will be suitable in terms of its business needs. The choice of a particular model depends on the type of organization and its business goals. Hence, it is imperative that an organization chooses an SPI model which closely represents its aims and goals and effectively uses it to launch its process improvement efforts. In order to evaluate the existing SPI models most effectively, a framework for selection is needed. Such a framework will help an organization to select the most suitable model for its software process improvement program. The objective of this research is to develop, define and justify the components of such a framework.

A framework that evaluates different SPI models can be used in an effective way by an organization to determine which SPI model best addresses the needs of that organization. The basic questions that an organization would look to be answered when it plans to choose a specific model could fall under a number of categories. The framework that has been created as part of this work attempts to answer these categories of questions. Questions fall under the different categories of *WHAT*, *WHICH*, *HOW* and *WHERE*. Important elements under each of these of four categories of questions include:

- *Goals of the model*: the base structure and goal of each model.
- *Underlying structure*: the base structure of the model.
- *Management role*: the role of management and the degree of commitment needed by each model.
- *Use of metrics*: the kind of metrics used by the model.
- *Benefits of model*: the benefits of compliance.
- *Underlying models*: model(s) and standards used for defining a particular model.
- *Rating scale*: the scale used to define the maturity level of the organization.
- *Organizational impact*: the process by which an organization uses a model to assess its capability and impacts after the assessment (i.e., how it is used for SPI).
- *Scope and domain*: which kind of industry has used the model.

The above model has been used to evaluate more than half a dozen SPI models, including CMM, ISO 9000-3, SPICE, Trillium, Bootstrap and AMI.

Acknowledgements

Partial support for this work was provided by the University Committee on Research, University of Nebraska at Omaha.

References

Brodman, J.G., Johnson, D.L., 1996. Return on investment from software process improvement as measured by US industry. Crosstalk, pp. 23–29.

Bush, M., 1991. Process assessments in NASA. In: Proceedings of the 13th International Conference on Software Engineering, pp. 299–304.

Butler, K.L., 1995. The economic benefits of software process improvement. Crosstalk 8 (7), 14–17.

Cheek, M., 1995. Using feedback to improve software development. IEEE Software 12 (3), 89–90.

Coallier, F., 1995. Trillium: A model for the assessment of telecom product development and support capability. Software Process Newsletter 2 (1), SPN3–SPN8.

Coletta, A., 1995. The SPICE project: An international standard for software process assessment, improvement and capability determination. In: Second Symposium on Software Quality Techniques and Acquisition Criteria, pp. 49–64.

DeMarco, T., 1982. Controlling Software Projects. Prentice-Hall, Englewood Cliffs, NJ.

Dewey, R.H., 1988. Software engineering technology and management. SPI International, Business Intelligence Program, Report No. 762.

Dion, R., 1992. Elements of a process improvement program. IEEE Software 9 (4), 83–85.

El-Emam, K., Drouin, J., Melo, W., 1998. SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. IEEE-CS, Silver Spring, MD.

Fenton, N.E., 1991. Software Metrics: A Rigorous Approach. Chapman & Hall, London.

Goldenson, D.R., Herbsleb, J.D., 1995. After the appraisal: A systematic survey of process improvement, its benefits and factors that inuence success. Technical Report CMU/SEI-95-TR-009, Software Engineering Institute.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., Zubrow, D., 1994. Benefits of CMM-based software process improvement: Initial results. Technical Report SEI-94-TR-13, Software Engineering Institute, Carnegie Mellon University.

Humphrey, W., 1989. Managing the Software Process. Addison–Wesley, Wokingham, UK.

Humphrey, W., 1992. Introduction to software process improvement. Technical Report CMU/SEI-92-TR-7, Software Engineering Institute.

Humphrey, W., Kitson, D., Kasse, T., 1989. The state of software engineering practice: A preliminary report. In: Proceedings of the 11th International Conference on Software Engineering, pp. 277–288.

Jones, C., 1992. Critical problems in software measurement, Version 1.00, Burlington, Mass.: Software Productivity Research (SPR).

Jones, C., 1996. The pragmatics of software process improvement. Software Process Newsletter 3 (5), SPN1–SPN4.

Kehoe, R., Jarvis, A., 1996. ISO 9000-3: A Tool for Software Product and Process Improvement. Springer, New York, NY.

Kitson, D., Humphrey, W.S., 1989. The role of assessment in software process improvement. Technical Report CMU/SEI-89-TR-3, Software Engineering Institute.

Koltun, P., 1992. Software process assessment: A first step to improvement. In Proceedings of the International Software Quality Exchange, Juran Institute, pp. 4B-31–4B39.

Kuvaja, P., Simila, J., Kranik, L., Bicego, A., Saukkonen, S., Koch, G., 1994. Software Process Assessment and Improvement: The Bootstrap Approach. Blackwell, Oxford.

Lebsanft, E., 1996. BOOTSTRAP: Experiences with Europe's software process assessment and improvement method. Software Process Newsletter 3(5), SPN6–SPN10.

Paulish, D.J., Carleton, A.D., 1994. Case studies of software-process-improvement measurement. IEEE Computer, Silver Spring, MD 27 (9), 50–57.

Paulk, M., Weber, C., Curtis, B., Chrissis, M., 1995. The Capability Maturity Model. Addison–Wesley, Wokingham, UK.

Pulford, K., Kuntzmann-Combelles, A., Shirlaw, S. A., 1995. Quantitative Approach to software Management: The AMI Handbook. Addison–Wesley, Wokingham, UK.

Zaharan, S., 1998. Software Process Improvement. Addison–Wesley, Wokingham, UK.

Dr. Hossein Saiedian is an associate professor of computer science at the University of Nebraska at Omaha. His primary area of research is software engineering and in particular models for quality software development, both technical and managerial ones. On the technical sides, he promotes the use of formal methods and is currently investigating the problems that have hindered the wide spread industrial use of such methods. On the managerial side, he is working on the development of a framework for the evaluation and application of software process improvement models, specially at local software organizations. Dr. Saiedian is also interested in object-oriented technology and its impact on the software development process. His other areas of interest include database management systems and computer science education. Dr. Saiedian is a member of the ACM, IEEE-CS, and on the editorial board of several journals, serves on the program committee of several national and international conferences, and chaired the 12th Conference on Software Engineering Education and Training, March 1999, New Orleans.

Kalyani Chennupati is a software engineer at the Metamor Information Technology Services, in Dallas, Texas.