

# An organizational maturity model of software product line engineering

Faheem Ahmed · Luiz Fernando Capretz

© Springer Science+Business Media, LLC 2009

**Abstract** Software product line engineering is an inter-disciplinary concept. It spans the dimensions of business, architecture, process, and the organization. Some of the potential benefits of this approach include cost reduction, improvements in product quality and a decrease in product development time. The increasing popularity of software product line engineering in the software industry necessitates a process maturity evaluation methodology. Accordingly, this paper presents an organizational maturity model of software product line engineering for evaluating the maturity of organizational dimension. The model assumes that organizational theories, behavior, and management play a critical role in the institutionalization of software product line engineering within an organization. Assessment questionnaires and a rating methodology comprise the framework of this model. The objective and design of the questionnaires are to collect information about the software product line engineering process from the dual perspectives of organizational behavior and management. Furthermore, we conducted two case studies and reported the assessment results using the organizational maturity model presented in this paper.

**Keywords** Software process assessment · Software engineering · Software process maturity · Software · Product Line · Software process improvement

## 1 Introduction

Many organizations that deal in broad areas of operation including consumer electronics, telecommunications, avionics, and information technology are using software product lines

---

F. Ahmed (✉)  
College of Information Technology, United Arab Emirates University,  
P. O. Box 17551, Al Ain, UAE  
e-mail: f.ahmed@uaeu.ac.ae

L. F. Capretz  
Department of Electrical & Computer Engineering, University of Western Ontario,  
London, ON N6A 5B9, Canada  
e-mail: lcapretz@eng.uwo.ca

because they effectively utilize their software assets. Clements et al. (2005) report that software product line engineering is a growing software engineering subdiscipline, and that organizations such as Philips<sup>®</sup>, Hewlett-Packard<sup>®</sup>, Nokia<sup>®</sup>, Raytheon<sup>®</sup>, and Cummins<sup>®</sup> are using it to achieve extraordinary gains in productivity, marketing time efficiency, and product quality. Clements (2001) defines the term, “software product line”, as a set of software-intensive systems sharing a common and managed set of features that satisfy the specific needs of a particular market segment or mission, and that are developed from a common set of core assets in a prescribed way. Some other terms for “software product line” that have been widely used in Europe include: “product families,” “product population,” and “system families”. The acronym BAPO (van der Linden 2002) (Business-Architecture-Process-Organization) defines process concerns associated with software product lines. The term, “Organization”, in BAPO is considered critical because it deals with the way an organization responds, adopts, and institutionalizes this concept.

The institutionalization of software product line engineering involves integrating or improving the business processes that are associated with the software product line infrastructure from the dual perspectives of organizational behavior and management. The entire institutionalization process involves an organizational level culture and a strong commitment to acquiring knowledge, skills and motivations to effectively initiate, launch and maintain a particular software product line. Furthermore, it requires that the concept becomes entrenched at all levels of the organization and is supported by the necessary infrastructure of organization-wide guidelines, required training, and necessary resources. Software product line engineering is considered to be institutionalized when it is widely accepted and integrated into the foundation of an organization. The successful institutionalization of a software product line within an organization has a profound impact on the product development behavior of that organization. It changes the mindset of the organization from single system development to multi-system software production. Organizational theory focuses on the design and structure of the organization dealing with software product line engineering. On the other hand, organizational behavior strives to understand the attitudes and performance of the employees. Finally, organizational management plays a vital role in institutionalizing the software product line within an organization because it creates and coordinates the infrastructure required. Thus, the adoption and implementation of organizational theory, behavior, and management are all required in the process of institutionalizing a software product line within an organization.

Research has been performed (Bosch 2000; Clements and Northrop 2002; Jazayeri et al. 2000; Weiss and Lai 1999) on software product line engineering processes that include software product line architecture, commonality and variability management, core asset management, and business case engineering as well as application and domain engineering etc. However, the organizational aspects of software product line engineering such as organizational structure, roles and responsibilities, organizational learning, change management, conflict management, organizational culture, organizational communication, and organizational commitment have not as yet been fully explored. The objective of this paper, then, is twofold: first, to provide an understanding of some of the key organizational factors affecting the process of institutionalizing software product line engineering within an organization; second, to suggest a comprehensive methodology for performing a maturity assessment of the organizational dimension of software product line engineering. Thus, the paper addresses a topic of immense importance from the perspective of software product line engineering.

1.1 Maturity models of software product line engineering process: the big picture

Software process maturity evaluation has been a key research area in the software engineering community because of its impact on the efficiency of the software product development process. Information about the level of maturity helps an organization understand its position in terms of process management and execution. Similarly, this information also helps organizations to introduce changes in the current process to make improvements, because a well-established and measureable process contributes markedly to the success of an organization. Software product lines are a relatively new concept in the history of software development and business. A lot of effort has been directed toward process methodology and the industrialization of this paradigm. The organizations dealing with software product lines also require a methodology to evaluate the maturity of the software product line process. Van der Linden et al. (2004) propose a four-dimensional software product line maturity evaluation framework based on the BAPO concept of operations that incorporates the business, architecture, process, and organizational aspects. This framework provides a preliminary foundation for a systematic and comprehensive strategy for a process maturity evaluation of software product line engineering. Figure 1 illustrates the conceptual layout of maturity evaluation approach proposed by van der Linden et al. (2004). The four dimensions of the framework include Business, Architecture, Process, and Organization. Furthermore, the proposed approach suggests the development of four separate maturity models corresponding to the four dimensions of BAPO. These theorists identify a maturity scale, which is comprised of five levels given in ascending order for each dimension of BAPO (See “Maturity Scales” column in Fig. 1). This classification method produces separate values for each of the four dimensions. However,

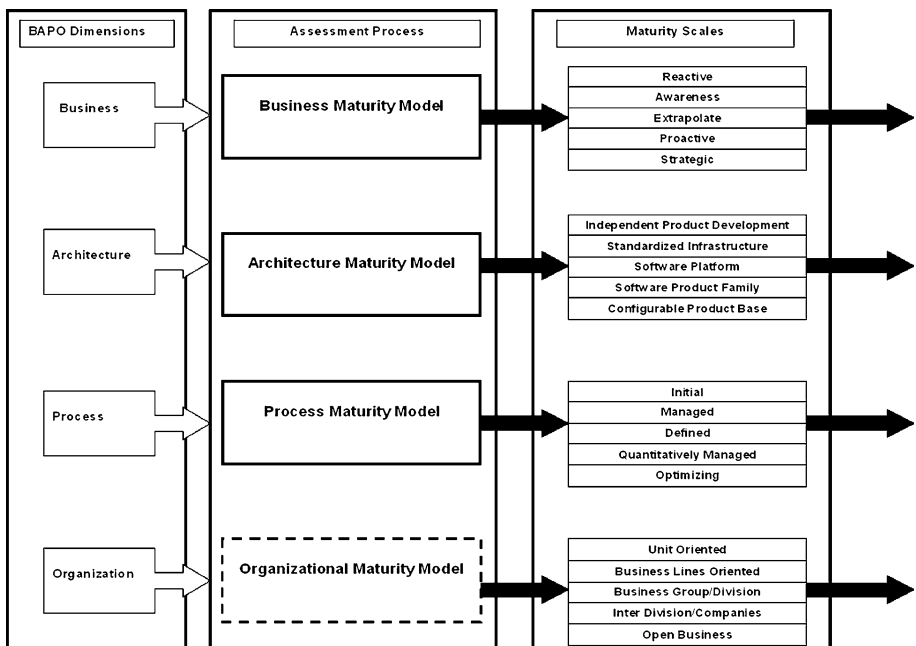


Fig. 1 van der Linden’s et al. software product line engineering maturity model

because of its relative novelty, the maturity models for each dimension have not yet been afforded much attention within the software engineering community.

This work presents an organizational maturity model for software product line engineering, and thus addresses one of the critical dimensions of the software product line process. The model provides a methodology for evaluating the current maturity of the organizational dimension of the software product line engineering process. To the best of our knowledge, this is the first study of its kind within the area of software product line engineering. As indicated by the highlighted rectangle in Fig. 1, this work concentrates solely on the organizational dimension, and thus, the other three dimensions of BAPO (business, architecture, and process) are beyond the scope of this study. The main objective of this research is to contribute toward a unified strategy for a process evaluation of software product line engineering.

## 1.2 Organizational dimension of software product line engineering: related work

Software engineering, in conjunction with the business management and organizational sciences, provides the foundation for the concept of software product line engineering. Thus, it has become an inter-disciplinary concept. Of these, the organizational dimension is the least addressed area in software product line engineering research due to its being a relatively new concept within software engineering paradigms. Instead, most of the research has been focused on the process, architecture, and business aspects of product line engineering. Nevertheless, some studies have presented new scenarios of organizational structure for software product lines. The researchers generally highlight the fact that a single domain engineering unit as well as several application engineering units is required from an organizational structure standpoint. Bosch (2001) presents four organizational models for software product lines: a development department in addition to business units, domain engineering units, and hierarchical domain engineering units. Bosch (2001) also illustrates a number of factors that influence the organizational model such as geographic distribution, project management maturity, organizational culture, and system types. Macala et al. (1996) report that a software product line demands careful strategic planning, a mature development process, and the ability to overcome organizational resistance. Dikel et al. (1997) share their experiences about initiating and maintaining software product lines at Nortel® and discuss organizational management and staffing issues. These issues are grouped into a set of six organizational principles, which Dikel et al. (1997) believe to be critical to the long-term success of a software product line. Jacobsen et al. (1997) focus on the roles and responsibilities of employees who deal with software product lines within an organization. Furthermore, Mannion (2002) elaborates on the management issues, organizational structure and culture, and learning patterns within the context of successfully adopting the concept of software product line engineering. Koh and Kim (2004) conclude that all members of an organization experience and share their own success stories about existing processes and organizational structures in order to successfully adopt the software product line approach. Clements and Northrop (2002) discuss organizational issues related to the software product line and identify four functional groups: the architecture group, the component-engineering group, the product line support group, and the product development group.

The organizational dimension of software product lines deals with the way an organization is able to handle complex relationships and multiple responsibilities (van der Linden et al. 2004). Toft et al. (2000) propose the “Owen Molecule Model,” which consists of three dimensions: organization, technology, and business. The organizational dimension of

this model deals with team hierarchy, individual roles, and operational models as well as with individual interaction and communication. Introducing software product line practice to an organization significantly affects the entire firm by fundamentally changing development practices, organizational structures, and the nature of task assignments (Birk et al. 2003). Bayer et al. (1999) have developed a methodology called PuLSE (Product Line Software Engineering) to enable the conception and deployment of software product lines within a large variety of enterprise contexts. PuLSE-BC is a technical component of the PuLSE methodology that provides ways to customize the PuLSE methodology to the specific needs of a particular organization. One of the support components of PuLSE entails an organizational aspect, which provides guidelines for developing and maintaining an organizational structure that is suitable for managing different product lines. According to Birk et al. (2003), introducing product line development to an organization can fundamentally change its development practices, its organizational structures, and its task assignments. These changes can, in turn, affect team collaboration and work satisfaction. Verlage and Kiesgen (2005) report a case study documenting the successful adoption of a particular software product line, and they conclude that organizational structure and management of change are significant issues.

An examination of related work reveals some key organizational factors. These include organizational structure, organizational culture, conflict management, change management, organizational commitment, communication, and organizational learning. In order to establish a foundation for the organizational maturity model of software product line engineering presented in this work, Ahmed et al. (2007) conducted and reported on an empirical investigation examining the effect of these key organizational factors on the performance of software product line engineering. This study covers a wide range of organizations actively involved in the process of software product line engineering. One of the objectives of this empirical experiment was to support the theoretical foundations of the organizational maturity model presented in this paper. We applied the key organizational factors derived from our literature review of software product line engineering and organizational theories as well as organizational management, and behavior in order to evaluate the maturity level of the organizational dimension of software product line engineering.

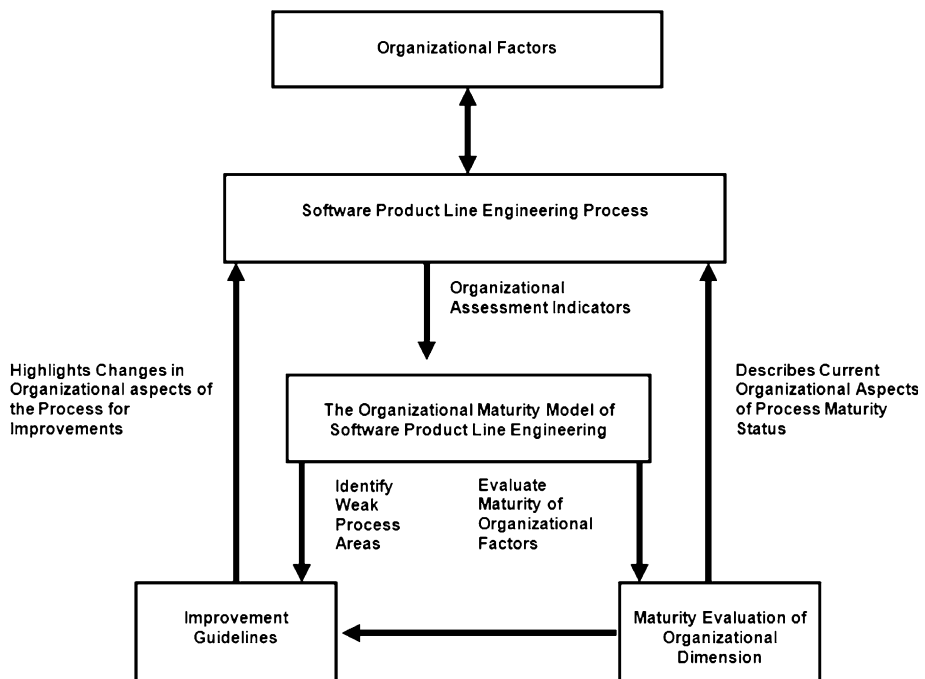
## **2 An organizational maturity model of software product line engineering**

Organizational theories facilitate an understanding of how to successfully integrate personnel into several types of engineering systems. Researchers from various fields such as anthropology, economics, management, political science, psychology, scientific management, sociology, and engineering have contributed toward the development of organizational theories. Because software engineering in conjunction with the business management and organizational sciences provides a suitable foundation for the concept of software product line engineering, it has become an inter-disciplinary concept. The organizational maturity model of software product line engineering aims to establish a comprehensive strategy for evaluating the organizational dimension. It describes an assessment methodology for determining the degree to which software product line engineering has become institutionalized. Furthermore, the model evaluates the organizational dimension's maturity in terms of how the various organizational factors dealing with software product line engineering are applied within the organization. The functional composition of the model consists of questionnaires purposely designed for evaluating maturity at each of the five levels proposed in this paper. Questionnaires have been

designed on the basis of related work performed in the organizational dimension of software product lines as well as organizational and management theories.

## 2.1 General scope of the organizational maturity model

A thorough assessment of the organizational dimension of software product line engineering is essential for improving the overall software product line engineering process within an organization. The overall objective of a maturity assessment model in software engineering is twofold. First, it provides a mechanism to perform assessments, and second, it provides guidelines to introduce further changes to the current process to make improvements. Like so many others, the software product line engineering process requires continual improvements over time. However, it is very difficult to develop an efficient and effective improvement plan unless it is based on the results of a prior comprehensive assessment exercise. Figure 2 illustrates the framework of such a comprehensive organizational assessment exercise for those organizations dealing with software product line development. The overall software product line engineering process involves the identification of many organizational factors that arise from organizational and management theories. These are blended into the software product line engineering process and can be used as indicators in the process assessment of the organizational dimension of software product line engineering. Hence, the product line engineering process either directly or indirectly applies these organizational theories. For example, if we examine a particular product line engineering process such as variability management, we see that this task requires communication between an established team and numerous others. The team thus



**Fig. 2** Framework of organizational maturity model of software product line engineering

formed requires a high degree of commitment and learning ability because there may be conceptual conflicts. Consequently, these organizational factors help to facilitate the product line engineering process. Similarly, a suitable organizational behavior and management activities plan will facilitate the execution of the product line engineering process. The organizational maturity model presented in this paper incorporates such factors in developing a comprehensive framework consisting of questionnaires and maturity level benchmarks to conduct the assessment. Furthermore, an organizational maturity assessment will determine the current status of the organizational dimension of the software product line engineering process within an organization. The assessment process will yield a number of recommendations for improvement based on the identification of weaknesses in the current process. Ideally, after the assessment process, the improvement guidelines will highlight the required changes in the current software product line engineering process to facilitate making improvements based on the assessment results. However, the maturity model presented in this work does not provide any guidelines for the improvement process, which we consider to be a future project arising from this study.

## 2.2 Configuration of the organizational maturity model

The functional configuration of the organizational maturity model consists of a set of two organizational dimensions: organizational behavior and organizational management. More specifically, these two dimensions are subdivided into seven organizational factors. Table 1 defines the hierarchy and domains of the organizational maturity model for software product line engineering. The organizational behavior dimension includes the factors of organizational culture, organizational commitment, and organizational learning. On the other hand, organizational management focuses on such issues as organizational structure, change management, conflict management, and organizational communication. In this paper, we use the term “organizational factor” to refer to a practice that facilitates a higher level of overall organizational performance and subsequently facilitates the process of institutionalizing software product line engineering within an organization. The term “organizational dimension” refers to a set of inter-related organizational factors that comprise the organizational dimension of software product line engineering. These organizational factors form the basis for the wording of the questionnaires that consist of several statements, which gauge the effectiveness of these factors in institutionalizing software product line engineering to evaluate the organizational dimension of the software product line engineering process. The following subsections elaborate on the theoretical aspects of these organizational factors.

**Table 1** Configuration of organizational maturity model

Organizational dimensions	Organizational factors
Organizational behavior	Organizational culture
	Organizational commitment
	Organizational learning
Organizational management	Organizational structure
	Change management
	Conflict management
	Organizational communication

### 2.2.1 Organizational behavior: literature review

Organizational culture has been characterized by many authors as comprising a set of shared values, beliefs, assumptions, and practices that shape and guide the attitudes and behaviors of individuals within the organization (Kotter and Heskett 1992; O'Reilly and Chatman 1996; Wilson 2001). And it has been largely associated with the people and the unique quality and style of the organization (Kilmann et al. 1985). Rosen (1995) acknowledges that the internal orientation of employees is based primarily on the culture, beliefs, ethics, and assumptions of that organization's staff, and, therefore, has the potential to be one of the most powerful influencing factors in strategic management. Moreover, organizational commitment is a work attitude that is directly related to the level of employee participation and to the intention to remain with the organization, and is, therefore, clearly linked to job performance (Mathieu and Zajac 1990). Crewson (1997) summarizes organizational commitment as being a combination of three distinct factors involving employee participation: first, a strong belief in and acceptance of the organization's goals and values; second, eagerness to work hard for the organization and third, desire to remain with the company. Argyris (1977) defines organizational learning as the process whereby the members of an organization respond to changes in the internal and external environment by detecting errors, which they subsequently correct in order to maintain the central features of the organization. Similarly, Marquardt and Reynolds (1994) define learning as a process by which individuals gain new knowledge and insights that change their behavior and actions. More specifically, Hames (1994) states that learning encompasses the acquisition and practice of new methodologies, skills, attitudes, and values necessary to thrive in a changing world. Lyles (1994) observes that organizations learn from their experiences and build on past experience that may influence future actions. Overall, organizational learning involves both the employees and the organization itself.

*Software product line engineering and organizational behavior* Some of the key process activities of software product line engineering, including domain engineering, software product line requirements engineering, commonality and variability management, and business case engineering etc., require a considerable amount of team effort, group discussion, and innovation. Software product line engineering requires a culture of openness where employees have the opportunity to participate in discussions and have the ability to express their views. For example, variability management is one of the critical process elements that require the active involvement of various parts of the organization, such as the business unit and the development unit, to identify specific areas for expansion in the product line architecture and introduce product-specific functionalities. Organizational commitment concerns the willingness of individuals and groups to achieve the long-term strategic objectives of an organization. The payback period of software product line engineering is relatively long compared to that of the single product development approach. Consequently, this transitional period requires a strong commitment from individuals, groups, and management to adopt the software product line engineering concept and to exercise patience with its development process. In software product line engineering, organizational learning can be classified into two domains: external and internal. External learning involves the necessary knowledge about customers, competitors, external environments, and market segments. This knowledge is necessary in order to effectively utilize the product line by exploiting product characteristics. In domain engineering, product line requirements and business case engineering etc., require that the organization have both established procedures and a means to acquire additional external learning. Overall, this type of learning helps an organization capture a major market share.



Internal learning, on the other hand, requires acquiring, transferring, and sharing a software product line methodology as well as ideas for process improvement and an understanding of the cross-functional requirements of product lines for individuals, groups, and the organization.

### 2.2.2 Organizational management literature review

An organization is comprised of the planned coordination of activities of a number of individuals for the achievement of some common, explicit purpose or goal, through a division of labor and function, and through a hierarchy of authority and responsibility (Schein 1988). Wilson and Rosenfeld (1990) define organizational structure as the established pattern of relationships between the parts of an organization, outlining communication as well as control and authority. According to Gordon (2002), organizational structure refers to the delineation of jobs and reporting relationships in an organization and coordinates the work behavior of employees in accomplishing the organization's goals. Beckhard and Harris (1987) consider organizational change to be an organization's movement from its present state to some future or target state. Furthermore, Todd (1999) defines change management as a structured and systematic approach, which provides a conceptual framework that encompasses strategy, politics, people, and process. Cao et al. (2000) observe that organizational change shows the diversity of an organization and illustrates the integration of technical and human activities that have inter-related functions within the organization. Walls and Callister (1995) maintain that conflict is a process in which one party perceives that its interests are being opposed or negatively affected by another party. Conflict management consists of diagnostic processes, inter-personal styles, negotiating strategies, and other interventions that are designed to avoid unnecessary conflict and reduce or resolve excessive conflict (Kottler 1994). Hellriegel et al. (1998) introduce four basic forms of conflict in an organization: goal-oriented, cognitive, affective, and procedural. Moreover, Jehn (1995) distinguishes between two kinds of intra-group conflict: task conflict and relationship conflict. Task conflict involves differences in viewpoints, ideas, and opinions, whereas relationship conflict is a perception of inter-personal incompatibility and includes annoyance and animosity among individuals (Medina et al. 2005). Communication is a central mechanism of interacting entities within the organization. White (1982) observes that communication is one of the vital processes that ensure that an organization has a healthy life span. The day to day operation of the company requires proper channels of communication. Communication is regarded as a powerful factor in organizational performance, because it also transmits the vision and objectives of the management. Witherspoon (1997) observes that communication is not only an essential aspect of these recent organizational changes, but effective communication can be seen as the foundation of modern organizations.

*Software product line engineering and organizational management* The theoretical foundations of software product line engineering divide the overall engineering process into two broad areas, application and domain engineering, and require stronger coordination and communication links between them. Verlage and Kiesgen (2005) present a case study documenting the successful implementation of software product lines in their organization. As a result, they report that the roles and the mapping of the roles related to the processes are not fixed; rather, they are interchangeable, or more precisely, dynamic. The organizations that have well-defined structures, which incorporate clearly identified roles for individuals and have strong coordination and communication links, are more likely to institutionalize a software product line than organizations that lack the supporting

coordination and communication. The process of evolving from single product development to a line of products necessitates a significant change within an organization. For example, in the case of requirements engineering, an organization must deal with product-specific requirements engineering as well as product line specific requirements engineering. The product-specific requirements engineering involves identifying the variability among products, whereas product line requirements engineering entails detecting the commonalities among different products. Furthermore, there is a need to introduce trade-off analyses for commonality and variability management. Introducing a new practice such as the development of a particular product line is relatively difficult within the existing setup of an organization, especially if it is not accompanied by appropriate modifications to the management plan. The successful implementation of any policy, particularly one involving a software product line, depends on how organizations manage their personal and organizational conflicts. All conflicts should be resolved in a professional environment that has a minimal impact on morale and productivity. At the same time, purposeful, task-related conflicts are very important for the growth of the organization. Any company that stifles healthy debate and professional conflict may find itself becoming stagnant and unproductive.

### 2.3 Framework of the organizational maturity model

Software process assessment models such as the CMM (Paulk et al. 1993) and BOOTSTRAP (Kuvaja et al. 1994) utilize the concept of staging in defining maturity levels, which illustrate qualitatively the maturity of the software engineering process. Similarly, the framework for the organizational maturity model presented in this paper also uses the approach of staging. In their work, van der Linden et al. (2004) define five maturity scales for assessing the organizational dimension of a software product line. These scales, in ascending order, are “unit oriented”, “business lines orientated”, “business group/division”, “inter-division/companies”, and “open business”. However, these scales linguistically describe the maturity of a company only in terms of its organizational structure. Expanding on their research, this paper proposes to develop a scale that reflects the maturity of a software product line in terms of an organizational dimension. Therefore, this proposed maturity scale will also include other organizational factors such as corporate culture, employee commitment, learning capacity, change management, conflict management, and communication level. The five levels in this maturity scale, in ascending order, are “Preliminary”, “Consistent”, “Streamlined”, “Matured”, and “Institutionalized”. Consequently, this maturity scale serves as a framework consisting of a set of questionnaires for each level of the scale. The set of statements in the questionnaires are divided into seven organizational factors and covers the two organizational dimensions. An organization’s maturity level is determined from the extent of agreement with each statement in the questionnaires.

Table 2 illustrates the framework of the organizational maturity model. Each maturity level consists of a set of statements and incorporates all seven organizational factors used in this study to shape the design of the questionnaires. The number of statements varies for each maturity level as well as for each organizational factor. For the remainder of this paper, the following set of abbreviations will be used: Organizational Culture (OC), Organizational Commitment (OCT), Organizational Learning (OL), Organizational Structure (OS), Change Management (CM), Conflict Management (CTM), and Organizational Communication (OCN). The following subsections describe the characteristics of an organization dealing with software product line engineering at a particular maturity

**Table 2** Framework of organizational maturity model

Maturity level	Organizational factors and number of statements in assessment questionnaires							
	OC	OCT	OL	OS	CM	CTM	OCN	Total
Preliminary	3	2	2	2	2	3	2	16
Consistent	4	2	2	3	4	2	2	19
Streamlined	3	3	4	3	4	3	3	23
Matured	5	3	3	3	4	3	3	24
Institutionalized	3	3	3	4	3	2	1	19

level of organizational dimension. In the questionnaires used with this maturity model, the following symbols and abbreviations are used.

- OF Organizational factor
- X Maturity level (an integer)
- Y Organizational factor number (an integer)
- S Statement
- I Maturity level (an integer)
- J Organizational factor number (an integer)
- K Statement number (an integer)

### 2.3.1 Preliminary (Level-1)

The preliminary stage of organizational maturity refers to an organization that does not as yet have a stable and organized environment for software product line engineering. In this case, there is no evidence that the organization performs software product line engineering activities in a coordinated way. The organization tends to carry out multiple product development only as a reaction to market demands, and the reusability of software assets is random and/or on an as-needed basis. There are currently no defined procedures to switch from a single product to a line of products. In regard to a software product line, there is a lack of strategic planning and an absence of business vision. Also, unnecessary task-related and personal conflicts are caused by a lack of understanding of the software product line engineering methodology. Similarly, communication among departments and its subunits is poor. Although organizations have a growing interest in developing a suitable infrastructure for software product line engineering, they lack the technical resources and skills to do so. The following measuring instrument uses seven organizational factors to illustrate the software product line organizational maturity of a company when it is at the “Preliminary” level.

#### OF.1.1 Organizational Culture

- S.1.1.1 The organization does not support the reusability of software assets and does not maintain them for any possible future use.
- S.1.1.2 The employees are not encouraged to generate new ideas and innovations for the software development methodology.
- S.1.1.3 New employees generally have difficulties adapting to the working environment of the organization.

### **OF.1.2 Organizational Commitments**

- S.1.2.1 Software product line engineering is not part of a strategic plan for the organization.
- S.1.2.2 Employees generally do not stay with the organization for long periods of time.

### **OF.1.3 Organizational Learning**

- S.1.3.1 The organization has not yet acquired sufficient knowledge about software product line engineering.
- S.1.3.2 The organization has not yet scheduled formal training sessions for the employees to learn about software product line engineering.

### **OF.1.4 Organizational Structure**

- S.1.4.1 The organization lacks a structure to support the software product line engineering process.
- S.1.4.2 Most of the departments in the organization work independently.

### **OF.1.5 Change Management**

- S.1.5.1 The organization has not yet defined policies regarding the introduction of software product line engineering.
- S.1.5.2 There is continual resistance against the adoption of a software product line engineering concept.

### **OF.1.6 Conflict Management**

- S.1.6.1 The organization has no specific protocol to deal with conflicts.
- S.1.6.2 Most of the conflicts in the organization are dealt with at a personal level.
- S.1.6.3 Personal conflicts are a major obstacle to the progress of the organization.

### **OF.1.7 Organizational Communication**

- S.1.7.1 Employees do not generally share their ideas or their work.
- S.1.7.2 There is a lack of communication among employees and individual departments as well.

## *2.3.2 Consistent (Level-2)*

The next organizational maturity level for the establishment of software product lines is Level-2 and is defined as “Consistent”. The organizations at this level are generally aware of the potential benefits of software product line engineering and encourage employees to acquire and share knowledge and skills. At the early stage of Level-2, the organization introduces changes in the structure of the company to support software product line engineering. The organization demonstrates actions and commitments to incorporate software product lines into its strategic plans and future directions. It has a defined policy to handle task conflicts, which mainly occur due to a misunderstanding of the software

process methodology. Organizational learning reveals an interest in the software product line engineering concept. Consequently, management supports employees in reusing software assets. A defined communication protocol among various entities within the organization is present and assists in the decision making. Overall, the organization understands the importance of software product line engineering in achieving business goals and is currently engaged in establishing an infrastructure to support software product lines. The following measuring instrument illustrates the set of statements that must be satisfied for an organization to achieve this level.

### **OF.2.1 Organizational Culture**

- S.2.1.1 The organization promotes new ideas and innovations in the software development methodology and takes the initiative to experiment with new concepts.
- S.2.1.2 Management and developers pay close attention to the reusability of software assets, and these assets are maintained and updated.
- S.2.1.3 Employees perceive themselves as participating in the decision-making process of the organization.
- S.2.1.4 The organization views its employees as a valued resource and an important contributor to its success.

### **OF.2.2 Organizational Commitments**

- S.2.2.1 Software product line engineering is considered as an option in the strategic plans of the organization.
- S.2.2.2 On average, employees remain in the organization for more than 3 years.

### **OF.2.3 Organizational Learning**

- S.2.3.1 Employees share knowledge and experience by having discussions with each other.
- S.2.3.2 Formal training periods are scheduled for the acquisition of knowledge about software product line engineering.

### **OF.2.4 Organizational Structure**

- S.2.4.1 The organization has established domain and application engineering units for the software product line engineering process.
- S.2.4.2 The various units within the organization work in a collaborative way and provide feedback to each other on the software product line engineering process.
- S.2.4.3 Software product line core assets are maintained at departmental levels, and the information is shared on a need-to-know basis.

### **OF.2.5 Change Management**

- S.2.5.1 Most of the software product line changes introduced to the organization are ad hoc and on an as-needed basis.
- S.2.5.2 Business plans are reviewed regularly and updated based on market and customer orientation.

- S.2.5.3 Management introduces changes in their strategies in response to the changing business environment.
- S.2.5.4 The changes in individual product functionalities are documented.

### **OF.2.6 Conflict Management**

- S.2.6.1 The organization has established a policy to address internal conflicts.
- S.2.6.2 Task-related conflicts concerning software product line engineering are handled by open discussion and the sharing of each other's viewpoints.

### **OF.2.7 Organizational Communication**

- S.2.7.1 The organization has established a defined communication protocol among its various entities.
- S.2.7.2 Information related to the software core assets is communicated to the personnel involved in product development.

#### *2.3.3 Streamlined (Level-3)*

An organization at Level-3, which is also referred to as “Streamlined,” is able to establish an infrastructure of software product lines by incorporating defined policies and procedures. The company’s strategic plans show the organization’s commitment to a long-term investment in software product line engineering. The employees are equipped with the required knowledge of the software product line methodology, and they, along with management, are continually learning to improve the process. Also, the employees understand the significance of the software product line in achieving the business goals of the organization. Moreover, the organizational structure completely defines the roles and responsibilities of individuals and groups in carrying out software product line tasks. Their experience in establishing a particular software product line allows them to introduce improvements to the organizational structure. Also, the changes in process methodology and product functionality are well documented and traceable. The organization’s communication framework is based on sharing information and providing feedback across departments. Software assets are maintained, used, and updated, which depicts a strong organizational culture toward the reusability of software assets. Management supports positive conflicts and encourages open discussions and the exchange of opinions to resolve task-related conflicts. Overall, the organization understands the process methodology of software product lines and is able to streamline activities of software product line engineering from the organizational perspective. The following measuring instrument illustrates the set of statements designed for Level-3.

### **OF.3.1 Organizational Culture**

- S.3.1.1 The workforce understands and is committed to the vision, values, and goals of the organization.
- S.3.1.2 The organizational culture supports the reusability of software assets and considers software product line engineering to be an important tool to achieve the strategic objectives of the organization.

- S.3.1.3 Employees are empowered to make appropriate decisions regarding their job execution.

### **OF.3.2 Organizational Commitments**

- S.3.2.1 Over the past 3 years, the organization as an entity has been steadily moving toward a software product line engineering approach in order to achieve its strategic objectives.
- S.3.2.2 The strategic plans define how an organization will achieve the technological capability to successfully adopt the concept of software product line engineering.
- S.3.2.3 Strategic planning allocates resources for software product line development.

### **OF.3.3 Organizational Learning**

- S.3.3.1 Necessary training for software product line engineering is provided to employees.
- S.3.3.2 The employees have opportunities to participate in problem solving and idea generation activities for software product line engineering.
- S.3.3.3 Employees have access to information from external resources, and management encourages experimenting with such knowledge to improve the software product line engineering process.
- S.3.3.4 The organization encourages employees to join formal and informal discussion forums for software product line engineering outside the organization.

### **OF.3.4 Organizational Structure**

- S.3.4.1 The roles and responsibilities of individuals and groups in such aspects as domain and application engineering are well defined and documented within the organization.
- S.3.4.2 A well-defined organizational unit with a clear set of guidelines handles the management of core assets with respect to the software product line.
- S.3.4.3 The organization tends to make changes in structure based on its experience in initiating and maintaining the software product line, and the organizational structure is in an evolutionary phase.

### **OF.3.5 Change Management**

- S.3.5.1 The organization has a well-defined change management plan to switch from single product development to a product line.
- S.3.5.2 A multi-dimensional management system tracks changes introduced at the software product line level and at the individual product level.
- S.3.5.3 The resistance to change in the organization is gradually decreasing.
- S.3.5.4 The change management plan is well communicated to all employees of the organization.

### **OF.3.6 Conflict Management**

- S.3.6.1 Management supports positive and constructive conflicts.
- S.3.6.2 Task conflicts do not affect the performance of individuals or departments. Rather, they facilitate the understanding of software product line implementation issues.
- S.3.6.3 Software product line variability and commonality management issues are discussed across departmental boundaries.

### **OF.3.7 Organizational Communication**

- S.3.7.1 Information about the core asset repository of the software product line is shared across departmental boundaries.
- S.3.7.2 The scope of the software product line is well communicated to developers and business analysts.
- S.3.7.3 The information and decisions related to business planning, product development, and marketing strategy as well as to customer and market orientation are shared and distributed across departmental boundaries.

#### *2.3.4 Matured (Level-4)*

The organizational maturity of software product line engineering at Level-4 is referred to as “Matured”. The organization at this level is able to coordinate organizational strategies with software product line engineering. Here, software product line engineering decisions are influenced by the shared visions of employees. Innovative measures are also introduced in software product line engineering, which depict the richness of the organizational culture in adopting product line engineering. Employees consider the software product line to be a vital strategic objective in achieving long-term organizational business goals. The organization keeps track of its mistakes and learns from experience to avoid repeating those mistakes. Personal conflicts are rare, and the number of task conflicts is minimal. Employees have access to required information to assist with their job performance. Open communication channels are present within the organization, and employees feel comfortable with expressing their points of view. Also, the process activities among various departments and subunits are synchronized. The following measuring instrument illustrates the set of statements designed for Level-4.

### **OF.4.1 Organizational Culture**

- S.4.1.1 Management supports both reactive and proactive innovations in the software product line process.
- S.4.1.2 Employees understand the importance of the software product line to the business vision and feel that the organization can realistically achieve its objectives.
- S.4.1.3 Employees generally perceive that the higher levels of management are helpful and supportive.
- S.4.1.4 Employees are encouraged to work in inter-disciplinary teams across departmental boundaries to share, disseminate, and acquire knowledge about software product line methodology.



- S.4.1.5 Employees understand the significance of the software product line in achieving the organization's business objectives, and they are encouraged to work together in teams and to care about the quantity and quality of the organization's outputs.

#### **OF.4.2 Organizational Commitments**

- S.4.2.1 Employees feel a sense of ownership for this organization.
- S.4.2.2 Employees consider the software product line to be a vital entity to achieve its long-term goals.
- S.4.2.3 The software product line is aligned with the strategic plans of the organization.

#### **OF.4.3 Organizational Learning**

- S.4.3.1 Formal and informal mechanisms are used to disseminate learning and knowledge within the organization.
- S.4.3.2 The organization learns from its experience and avoids repeating its mistakes.
- S.4.3.3 The innovations to the software product line are aligned with the existing business goals of the organization.

#### **OF.4.4 Organizational Structure**

- S.4.4.1 The organizational structure supports the software product line concept and the roles, and the mapping of the roles with respect to the processes are not fixed. Rather, they are dynamically interchangeable, strongly coordinated, and well documented.
- S.4.4.2 A well-established business unit oversees its marketing plans and strategies in coordination with its domain and application engineering units.
- S.4.4.3 A joint team from the domain and application engineering units oversees the synchronization of activities in both departments.

#### **OF.4.5 Change Management**

- S.4.5.1 The changes to institutionalize the software product line are well accepted by the employees.
- S.4.5.2 The organizational change management plans define how an organization will achieve the technological capability to successfully adopt the concept of the software product line.
- S.4.5.3 The employees understand the importance and impact of change with respect to the business of the organization.
- S.4.5.4 Any changes made to product functionalities and to the product line are well communicated to developers and are approved by a defined chain of command.

#### **OF.4.6 Conflict Management**

- S.4.6.1 The number of personal conflicts is gradually declining.
- S.4.6.2 The organization is experiencing a reduced number of task conflicts every year.
- S.4.6.3 Employees believe that task conflicts help to improve the software product line methodology, and they participate actively in discussions.

### **OF.4.7 Organizational Communication**

- S.4.7.1 A strong and open communication channel among various entities of the organization is present.
- S.4.7.2 The organization has an established inter-communication protocol among external and internal entities for the dissemination of market intelligence.
- S.4.7.3 Employees feel that they have the information they need to successfully perform their duties.

#### *2.3.5 Institutionalized (Level-5)*

The highest organizational maturity level is referred to as “Institutionalized”. An organization at Level-5 considers the software product line to be a strategic asset that can be mobilized to achieve the desired business objectives. The software product line plays an important role in the business vision of the organization. Research and development in the product line process methodology is a continuing process. The organization learns from its previous experience and mistakes, and they use those lessons to improve the process methodology. Employees prefer to work in teams, and intra-team trust plays a significant role in minimizing task conflicts. Information flows within the organization are free and smooth. Management listens closely to the employees and believes that quality performance and productivity can only be achieved by openly introducing the recommended improvement plans to employees. Also, the management allows employees to experiment with their ideas and innovative concepts. All of the organizational units work collectively, and everyone believes that the organization can realistically achieve its objectives. The following measuring instrument illustrates the set of statements designed for Level-5.

#### **OF.5.1 Organizational Culture**

- S.5.1.1 The organizational culture supports innovations in the software product line and has successfully introduced improvements in its process methodology.
- S.5.1.2 Employees perceive that everyone within the organization places a strong emphasis on the quality, service, and reliability of its products.
- S.5.1.3 Employees are seen as being a key source of ideas for improvements in quality and productivity.

#### **OF.5.2 Organizational Commitments**

- S.5.2.1 The organizational strategic planning includes consideration of the software product lines as being important and even strategic assets.
- S.5.2.2 The software product line is an integral part of the organization’s business vision.
- S.5.2.3 Employees share a high degree of commitment to make the organization’s strategic vision a reality.

#### **OF.5.3 Organizational Learning**

- S.5.3.1 Research and development in the software product line is a continuous process within the organization.
- S.5.3.2 The organization is committed to learning and improving the level of knowledge in the area of software product lines.

S.5.3.3 The organization has successfully employed innovations in the software product line development.

#### **OF.5.4 Organizational Structure**

S.5.4.1 The organizational structure supports the strategic plans of the company.

S.5.4.2 Various subunits within the organization are actively encouraged to operate in a coordinated way by cooperating effectively to achieve the overall organizational objectives.

S.5.4.3 Cross-functional teams are established to oversee the whole software product line process and to support management decision making.

S.5.4.4 The cross-functional activities in various departments are synchronized.

#### **OF.5.5 Change Management**

S.5.5.1 The organization regularly conducts reviews of the changes made in the software product line process methodology.

S.5.5.2 The organization learns from the changes and understands their impact on the performance of the organization in managing the software product line.

S.5.5.3 Organizational changes are within the scope of the software product line and cover the requirements of the targeted market segment.

#### **OF.5.6 Conflict Management**

S.5.6.1 Task conflicts help to create improvement plans for the software product line methodology.

S.5.6.2 Intra-group trust plays a critical role in managing conflicts within the organization.

#### **OF.5.7 Organizational Communication**

S.5.7.1 Communication is perceived as flowing freely and accurately throughout the organization.

### 2.4 Performance scale

The maturity level of an organization defines its ability to institutionalize software product line engineering. The ratings described in Table 3: “Completely Agree (4)”, “Largely Agree (3)”, “Partially Agree (2)”, and “Do Not Agree (1)” are used to measure each organizational factor. This rating threshold provides a set of quantitative measurements that reflect the organization’s agreement with each statement in the questionnaire. The scale point of 0, expressed as “Doesn’t Apply,” is designed to increase the flexibility of the model, and consequently, it is equivalent to a 4 in the rating algorithm. Our performance scales and our threshold values are very similar to those used in the BOOTSTRAP (Wang and King 2000) methodology, as illustrated in Table 3. We intentionally based our scale on a previously existing approach in order to maintain consistency between our assessment and existing rating scales, which were already in use and which have been validated and well accepted by the software industry. However, despite these similarities, we have

**Table 3** Performance scale

Scale	Linguistic expression of performance scale	Linguistic expression of BOOTSTRAP	Rating threshold (%)
4	Completely agree	Completely satisfied	$\geq 80$
3	Largely agree	Largely satisfied	66.7–79.9
2	Partially agree	Partially satisfied	33.3–66.6
1	Does not agree	Absent/poor	$\leq 33.2$
0	Does not apply	Does not apply	–

introduced some changes in the linguistic expressions of the performance scales. The major reason for these changes relates to the current design of the questionnaires. Specifically, our questionnaires take the self-assessment approach into account where an organization is able to evaluate the maturity of their organizational dimension by expressing their own level of agreement with the statements.

## 2.5 Rating method

The rating method adopted in this organizational maturity model derives its foundations partially from the BOOTSTRAP algorithm (Wang and King 2000) of software process assessment. However, the structure of the rating method uses different terminologies, such as Performance Rating ( $PR_{OF}$ ), Number of Agreed Statements ( $NA_{OF}$ ), Pass Threshold ( $PT_{OF}$ ), and Organizational Maturity Level (OML). The following section describes each terminology in detail.

Let  $PR_{OF}[I, J]$  be a rating of the  $I$ th organizational factor of the  $J$ th maturity level. Then, using the performance scale defined in Table 3,  $PR_{OF}[I, J]$  can be rated as

$$\begin{aligned}
 PR_{OF}[I, J] &= 4, \text{ if the extent of agreement with the statement is at least } 80\% . \\
 &= 3, \text{ if the extent of agreement with the statement is between } 66.7 - 79.9\% . \\
 &= 2, \text{ if the extent of agreement with the statement is between } 33.3 - 66.6\% . \\
 &= 1, \text{ if the extent of agreement with the statement is less than or equal to } 33.2\% . \\
 &= 0, \text{ if the statement does not apply to this assessment.}
 \end{aligned}$$

An  $I$ th statement at the  $J$ th maturity level is considered to agree if  $PR_{OF}[I, J] \geq 3$  or  $PR_{OF}[I, J]$  is equal to 0. If the number of statements to which people agree at maturity level “ $J$ ” is  $NA_{OF}[J]$ , then it is defined by the expression

$$\begin{aligned}
 NA_{OF}[J] &= \text{Number of } \{PR_{OF}[I, J] \mid \text{Agreed}\} \\
 &= \text{Number of } \{PR_{OF}[I, J] \mid PR_{BP}[I, J] \geq 3 \text{ or } PR_{OF}[I, J] = 0\}.
 \end{aligned}$$

Table 4 illustrates the pass threshold of 80% at each maturity level. The values are calculated to the nearest hundred. The maturity level is considered as a pass if people agree to 80% of the statements in the questionnaire. If  $N_{OF}[J]$  is the total number of statements at the  $J$ th maturity level, then the pass threshold ( $PT_{OF}$ ) at the  $J$ th maturity level is defined as

$$PT_{OF}[J] = N_{OF}[J] \times 80\%.$$

The Organizational Maturity Level (OML) is defined as the highest maturity level at which the number of statements agreed upon is more than or equal to the pass threshold ( $PT_{OF}[J]$ ), given by

**Table 4** Rating threshold

Maturity level	Total statements	Pass threshold 80%
Preliminary	16	13
Consistent	19	15
Streamlined	23	18
Matured	24	19
Institutionalized	19	15

$$\text{OML} = \max\{J | \text{NA}_{\text{OF}}[J] \geq \text{PT}_{\text{OF}}[J]\}.$$

### 3 Reliability and validity analysis of questionnaires

The two most important aspects of precision in questionnaire-based assessments are reliability and validity. Reliability refers to the reproducibility of a measurement, whereas validity refers to the agreement between the measured value and the true value. The potential threats to the external validity of questionnaire-based assessments are the reliability and validity of the measuring instrument. In order to test these two concepts, we conducted a pilot study that required organizations to show the extent of their agreement with each statement in the questionnaires. Nine organizations participated in this pilot study. The reliability of the questionnaires was evaluated by using the internal-consistency analysis method. Internal-consistency analysis was performed using the coefficient alpha (Cronbach 1951). Table 5 illustrates the results of the reliability analysis, in which the coefficient alpha ranges from 0.56 to 0.91. Nunnally and Bernste (1994) found that a reliability coefficient of 0.70 or higher was satisfactory. However, other reliability literature, such as (van de Ven and Ferry 1980), suggests that a reliability coefficient of 0.55 or higher was satisfactory, and (Osterhof 2001) concluded that 0.60 or higher was satisfactory. Accordingly, our analysis shows that most of the questionnaire items developed for this organizational maturity model satisfy the criteria stipulated in (Nunnally and Bernste 1994), whereas some of the items have an alpha coefficient of less than 0.70 but still fall within the acceptable ranges stated in (van de Ven and Ferry 1980) and (Osterhof 2001). Overall, our analysis shows that all of the questionnaire items are considered reliable, since each construct has an alpha of at least 0.55 or higher, therefore falling within the acceptable ranges.

**Table 5** Reliability analysis of organizational factors (coefficient alpha)

Maturity level	Organizational factors						
	OC	OCT	OL	OS	CM	CTM	OCN
Preliminary	0.59	0.63	0.70	0.78	0.88	0.61	0.70
Consistent	0.72	0.88	0.85	0.61	0.91	0.77	0.70
Streamlined	0.89	0.85	0.92	0.89	0.82	0.82	0.85
Matured	0.80	0.84	0.71	0.61	0.56	0.65	0.60
Institutionalized	0.81	0.76	0.69	0.86	0.84	0.72	*

\* The construct has only one item of evaluation; hence the calculation of the coefficient alpha is not possible

**Table 6** Construct validity of organizational factors (PCA)

Maturity level	Organizational factors						
	OC	OCT	OL	OS	CM	CTM	OCN
Preliminary	1.69	1.46	1.64	1.72	1.80	1.90**	1.54
Consistent	2.22**	1.80	1.75	1.88	3.20	2.24	2.36
Streamlined	2.50	2.32	3.44	2.45	2.75	2.24	2.36
Matured	2.93	2.42	2.05	1.70	1.79**	1.89	1.69
Institutionalized	2.23	2.09	1.99	2.90	2.27	1.57	*

\* The construct has only one item, so PCA is not possible. \*\* More than one factor has an eigenvalue >1.0

Construct validity, according to Campbell and Fiske (1959), occurs when the scale items in a given construct move in the same direction, and thus, are highly correlated. A principal component analysis (Comrey and Lee 1992) was performed and noted for all seven organizational factors at each maturity level. Table 6 provides a measure of construct validity. We used eigenvalues (Kaiser 1970) and scree plots (Cattell 1966) as reference points to observe construct validity using principal component analysis. Also, we used the eigenvalue-one criterion, also known as Kaiser criterion (Kaiser 1960; Stevens 1986), which means that any component having an eigenvalue greater than one is retained. Eigenvalue analysis reveals that most of the items in the questionnaires formed a single factor, whereas in some cases two components were formed, such as “Conflict Management” at Level-1, “Organizational Culture” at Level-2, and “Change Management” at Level-4. The eigenvalue for the second component in those cases is slightly higher than the threshold of 1.0. A cell value in Table 6 with “\*\*” represents cases where more than one factor is formed. The scree plots clearly show a cut-off at the first component. Therefore, the construct validity can be regarded as being sufficient. It is important to note that principal component analysis and internal consistency analysis require more than one item in a construct to calculate the eigenvalue and the coefficient alpha. The construct of “Organizational Communication” at Level-5 has only one item, and therefore, we are unable to perform a principal component analysis and calculate the coefficient alpha for it. The cells with a “\*” in Tables 5 and 6 highlight that construct. Overall, the measurements of reliability and validity show that the measurement procedures used in this organizational maturity model meet the acceptable level of psychometric properties.

#### 4 Case studies

In order to perform the organizational maturity assessment, we applied our model to two organizations currently involved in the process of software product line engineering. In order to protect the privacy of the two organizations, they will be referred to as “A” and “B”. These organizations are using the concept of software product line engineering for at least the last 3 years. The reason for using organizations with 3 years’ experience in the software product line as a criterion is the typically long-term payback period for software product line development. In order to enhance the external validity, we ensured that organizations had started to experience the benefits of a software product line in terms of either an immediate or a future payback. Organization “A” is a software development firm that has been in business for over a decade and has software development sites all over the globe. Organization “B”, one of the largest companies in the automotive industry and is

**Table 7** Details of assessment result of case study “A”

Preliminary level-1		Consistent level-2		Streamlined level-3		Matured level-4		Institutionalized level-5	
Statement #	Value	Statement #	Value	Statement #	Value	Statement #	Value	Statement #	Value
S1.1.1	1	S.2.1.1	3	S.3.1.1	4	S.4.1.1	3	S.5.1.1	3
S1.1.2	1	S.2.1.2	3	S.3.1.2	4	S.4.1.2	3	S.5.1.2	3
S1.1.3	1	S.2.1.3	3	S.3.1.3	4	S.4.1.3	3	S.5.1.3	3
S1.2.1	1	S.2.1.4	3	S.3.2.1	3	S.4.1.4	2	S.5.2.1	2
S1.2.2	1	S.2.2.1	2	S.3.2.2	3	S.4.1.5	2	S.5.2.2	2
S1.3.1	1	S.2.2.2	3	S.3.2.3	3	S.4.2.1	3	S.5.2.3	3
S1.3.2	1	S.2.3.1	3	S.3.3.1	3	S.4.2.2	3	S.5.3.1	2
S1.4.1	1	S.2.3.2	3	S.3.3.2	3	S.4.2.3	2	S.5.3.2	2
S1.4.2	1	S.2.4.1	3	S.3.3.3	4	S.4.3.1	2	S.5.3.3	2
S1.5.1	1	S.2.4.2	3	S.3.3.4	4	S.4.3.2	2	S.5.4.1	2
S1.5.2	1	S.2.4.3	2	S.3.4.1	3	S.4.3.3	3	S.5.4.2	2
S1.6.1	1	S.2.5.1	3	S.3.4.2	3	S.4.4.1	3	S.5.4.3	2
S1.6.2	1	S.2.5.2	3	S.3.4.3	3	S.4.4.2	3	S.5.4.4	2
S1.6.3	1	S.2.5.3	3	S.3.5.1	3	S.4.4.3	3	S.5.5.1	2
S1.7.1	1	S.2.5.4	2	S.3.5.2	3	S.4.5.1	3	S.5.5.2	3
S1.7.2	1	S.2.6.1	3	S.3.5.3	4	S.4.5.2	3	S.5.5.3	3
		S.2.6.2	3	S.3.5.4	3	S.4.5.3	2	S.5.6.1	3
		S.2.7.1	3	S.3.6.1	3	S.4.5.4	3	S.5.6.2	3
		S.2.7.2	3	S.3.6.2	3	S.4.6.1	2	S.5.7.1	3
				S.3.6.3	3	S.4.6.2	2		
				S.3.7.1	3	S.4.6.3	2		
				S.3.7.2	3	S.4.7.1	3		
				S.3.7.3	3	S.4.7.2	2		
						S.4.7.3	3		

using software product line engineering to develop embedded systems for producing various parts of the automobile. Both organization “A” and “B” are from North America. The organizations are assumed to be large scale organizations both having more than 3,000 employees working in various sections. Table 7 shows the detailed assessment results of Organization “A”. The numerical values entered in each cell of table 7 represent the organization’s agreement with the statements in the questionnaires at each maturity level. Table 8 shows a summary of the assessment results. It is important to note that, according to the rating method discussed in Sects. 2 and 2.4, a statement is considered to be agreed upon if the performance rating shown in Table 3 is either greater than or equal to 3, or at 0. Organization “A” is at the “Streamlined” maturity level, which is Level-3, while Organization “B” is at Level-4 or the “Matured” level. The following section elaborates on the assessment methodology used in this study.

**Table 8** Summary of assessment results of case studies

Maturity level	Total statements	Pass threshold 80%	Organization "A" $NA_{OF}$	Organization "B" $NA_{OF}$
Preliminary	16	13	0	0
Consistent	19	15	16	0
Streamlined	23	18	23	22
Matured	24	19	14	21
Institutionalized	19	15	9	10

$NA_{OF}$  total number of agreed statements

#### 4.1 Assessment methodology

- The two participating organizations are from North America. In terms of size, they are considered to be large since each has a total of over 3,000 employees working in various departments.
- In the first stage of the study, we established contact with individuals in the two organizations to request their participation in this study. Specifically, we sent personalized emails to the individuals describing the scope and objectives of the study. Since the individuals contacted were working in the area of software product line engineering, their answers were directly applicable to the study. Furthermore, we informed the participants that the assessment being conducted was part of a Ph.D. research project and that neither the identity of an individual nor of an organization would be disclosed in the resulting Ph.D. thesis or in any subsequent research publications.
- The questionnaires designed for this maturity model are used to measure the maturity of the organizational dimension of each company's software product line engineering. The individuals participating in the study were requested to provide the extent of their agreement with each statement by using the performance scale ranging from 0 to 4. This performance scale is illustrated in Table 3.
- Our assessment methodology uses a top-down approach, where the more emphasized characteristics can be identified by moving from a lower to a higher level in the questionnaire. Consequently, the respondents must complete the questionnaire by starting at Level-1 and finishing at Level-5.
- All of the participants in this study were volunteers, and no compensation of any form was offered or paid. We also told the respondents that if for any reason they did not want to answer any question, to please leave it blank.
- On average, the respondents to this study had been associated with their respective organizations for the last 3 years. The minimum educational qualification of the respondents was an undergraduate university degree and the maximum was a Ph.D. degree. Most of the respondents generally belonged to middle or senior technical management and were associated with the software development process. However, some of them were from other departments such as marketing, sales, and business development. Some of the participants had policy-making roles or were involved in implementing organizational strategies from the top to the bottom.
- We highlighted some major sources of data such as documents, plans, models, and actors for the participants in the study. This was done to reduce the likelihood of



inaccurate estimations from the respondents' and to insure the reliability of the approach.

- Our assessment was not conducted by the usual on-site method. Specifically, we neither visited the organizations in person nor had meetings with the individual respondents to discuss the questionnaires. Our major source of contact and communication with the respondents was via email.
- We received multiple responses from each organization and thus limited the amount of bias in the sample. A variety of respondents from each organization provided a more accurate overall description of the company. Also, we performed an inter-rater agreement analysis, which is described in following section. This analysis provided information regarding the extent of agreement among the raters within each organization.

#### 4.2 Post-assessment improvement activity: a preliminary study

The maturity model presented in this work does not provide specific guidelines for the improvement process. We have assigned this step to being a post-assessment activity, because we want to limit the scope of this work to the domain of maturity assessment and target the rest for future work. But, in order to better understand the use and significance of the organizational maturity model presented in this work, we have discussed and described the post-assessment improvement activities with the participants from one of the organizations under study. In order to gain a better understanding we used one of the organizational factors, "conflict management" in this preliminary study. Organization "A" was found to be at Level-3 i.e. "Streamlined". After the assessment, the organization made a number of observations regarding the management of conflicts in the organization such as the need for employee training to provide them with a more comprehensive understanding of conflict management approaches, so that employees could better differentiate between personal and task conflicts. This observation was based initially on the fact that a number of conflicts within the organization was not declining over time. They also observed a lack of mechanism for acknowledging and handling inter-departmental issues. There was also a lack of an appropriate mechanism for conflict recording and reporting. These observations were further elaborated on in terms of the software product line engineering process as follows: There was a lack of coordination between the software product line architecture and the business development teams because most of the issues concerned managing the variability among products. There was also a lack of understanding of the software product line requirements and how to manage those requirements across the product development in terms of obvious commonalities. In order to set improvement plans, the organization introduced a comprehensive training program to increase the employees' understanding of conflict management. The training involved imparting knowledge about management procedures and inter-personal skills with respect to indentifying, analyzing, and managing conflicts. The training also emphasized the differences between personal and task conflicts. Conflict resolution teams were established at the organizational level that included members from various departments to address the inter-departmental issues. The conflict resolution team also proposed the writing of a formal written procedure manual to record conflict. In addition to those steps, the organization established a joint review committee consisting of participants from the software product line architecture group and the business development group. The objective of this committee was to discuss the issues arising among those groups and to formulate a strategy to overcome those problems. The committee also recommended a loosely structured boundary for those departments in order to

increase the level of communication and gain a better understanding of the underlying processes of the two departments. It is important to mention here that we did not visit the organization in person to evaluate and/or record the improvements made by the organization. The preliminary study of the improvement process reported upon here is based on our discussions with the people in the organization.

### 4.3 Inter-rater agreement analysis

Since multiple respondents within a single organization may produce conflicting opinions about the various organizational factors within the same company, we performed and reported on an inter-rater agreement analysis. This analysis provided information about the extent of agreement among the raters within one organization. Inter-rater agreement corresponds to reproducibility in the evaluation of the same process and adheres to the same evaluation specification (Lee et al. 2001). According to El Emam (1999), the level of inter-rater agreement measures the extent of agreement among independent respondents regarding their ratings of the same software engineering practices. The Kendall coefficient of concordance ( $W$ ) (von Eye and Mun 2005) is often used to evaluate inter-rater agreement. Where ordinal data is used, it is preferred to such other methods as Cohen's Kappa (Cohen 1960). The symbol, " $W$ ", indexes the divergence of the actual agreement shown in the data from the ideal agreement. In our study, we conducted an inter-rater agreement analysis using Kendall and Kappa statistics. Table 9 reports the Kendall and Kappa statistics of Organization "A". Values of Kendall's  $W$  and the Fleiss Kappa coefficient can range from 0 to 1, with 0 indicating perfect disagreement and 1 indicating perfect agreement (Landis and Koch 1977). The benchmark for Kappa (El Emam 1999) includes four level scales (<0.44 poor, 0.44–0.62 moderate, 0.62–0.78 substantial, and >0.78 excellent). The Kappa coefficient observed in this study ranges from 0.65 to 0.77 and hence falls into the category of being substantial. The total number of respondents from Organization "A" and "B" were 18 and 21, respectively.

### 4.4 Limitations of the assessment methodology

Questionnaire-based maturity models are susceptible to certain limitations, which is the case with this model. Some of the limitations associated with this model of software product line engineering are as follows:

- The first limitation involves the degree of completeness of the model. Although we used seven different organizational factors, which were spread over five maturity

**Table 9** Inter-rater agreement analysis

Maturity level	Kendall statistics		Kappa statistics	
	Kendall's coefficient of concordance ( $W$ )	$\chi^2$	Fleiss kappa coefficient	$Z$
Level-1	0.80	61.30*	0.77	10.33*
Level-2	0.70	59.40*	0.67	9.01**
Level-3	0.69	48.39*	0.65	8.60*
Level-4	0.75	63.20*	0.71	10.02*
Level-5	0.77	55.22**	0.72	9.58**

\*  $P < 0.01$ ; \*\*  $P < 0.05$

levels, there may have been other factors that influenced the institutionalization process of software product lines. Other contributing factors not considered in this model included organization size, economic conditions, and political conditions.

- The second limitation of the methodology was the issue of subjective assessment. We used statistical techniques that were most commonly used in software engineering to ensure the reliability and validity of the questionnaire-based assessment approaches. However, our measurements were still largely based on the subjective assessment of individuals.
- Although we used multiple respondents within the same organization to reduce bias, bias is still a core issue in decision making and evaluating questionnaire-based responses. Product line engineering is a relatively new concept in software development, and not many of the organizations in the software industry have institutionalized and launched this concept. Hence, collecting data for determining the level of reliability and validity of the various assessment items from the software industry was a limitation.
- The degree of respondent participation also affected the accuracy of the results. As previously mentioned, we asked the respondents to consult major sources of relevant data in their organizations to reduce the possibility of inaccurate judgment when filling in questionnaires. However, the data collection was largely dependent on the individuals' efforts to obtain the required information before responding to the statements presented in the questionnaire.
- We did not ask the respondents to provide us with their confidence levels in answering the questions quantitatively, but we did request them to subjectively reply to the questions to the best of their abilities. In order to handle this particular aspect, we conducted and recorded the results of this inter-rater agreement analysis, while still assuming that this aspect was an additional limitation of the assessment methodology.
- Our assessment methodology did not account for the role of independent assessors even though their role is an important aspect of maturity assessment modeling. Their role defines the level of coordination between the assessor and the internal assessment team and provides for an evaluation. The current case studies are based on self-assessment.
- The methodology evaluates and quantifies the maturity level of the different organizational factors as well as gauges the overall maturity level of the organizational dimension. However, our maturity model does not provide any guidelines for an improvement process, which we consider to be a subsequent project emanating from this study.

Although the organizational maturity model presented in this paper has both some general and specific limitations, it still provides a comprehensive approach to evaluating the maturity level of the organizational dimension for software product line engineering. Furthermore, it provides a suitable foundation for future research in this area.

#### 4.5 Utilization of the organizational maturity assessment model

One of the advantages of using maturity models in software engineering is that they have the ability to obtain inside information about the current maturity level of the different process-related activities in a particular organization. Ideally, this information provides a basis for improvement plans and activities. Furthermore, maturity models are also advantageous to individual organizations because companies with high maturity ratings are more attractive to potential customers. We summarized the advantages of the

organizational maturity model from such perspectives as software engineering research, various organizational aspects, product development, and process improvements.

- Overall, the maturity level model presented in this work provides information that can be used to improve the organization's process methodology and complementary product development activities within the organization.
- The overall performance of an organization depends on a number of critical factors that facilitate the enhancement of the human resource component of a company. Since technologies and organizational formats are evolving rapidly, companies must monitor the factors affecting the performance of employees involved in product development. The monitoring of such organizational factors helps in achieving the company's ultimate goal of developing and profiting from its products. The maturity assessment model presented in this work will help companies monitor and evaluate the overall working environment of an organization.
- The model presented here also highlights a methodology for evaluating some of the organizational factors affecting the operation of a company. Moreover, such an evaluation can provide inside information on the factors requiring improvement by management. For example, if management discovered that organizational communication is at a lower maturity level then they could introduce changes to the communication protocols to improve it. Such improvements could subsequently help in the product development process, which is the ultimate goal of the organization.
- The software product line is gaining popularity and many organizations around the world are currently involved in applying this concept. Our model provides a preliminary conceptual framework for the maturity assessment of software product line engineering. Consequently, this area of study still requires future contributions from software engineering researchers.

## 5 Final remarks

This research contributes toward the establishment of a comprehensive and unified strategy for a process maturity evaluation of software product line engineering. It also presents an organizational maturity model for evaluating the "organizational dimension" of software product line engineering. Furthermore, it helps us to understand the institutionalizing process of software product line engineering within a company. The model comprises five maturity levels, a set of questionnaires for each of these five maturity levels, performance scales, and a comprehensive rating method. The responses to the statements in the questionnaires, when applied in conjunction with the rating methodology, provide a quantitative means of evaluating the maturity level of the organizational dimension of software product line engineering. The case studies conducted in this research reveal the maturity of software product lines in those organizations. This research reinforces the current perception that software product line engineering requires a comprehensive and precise alignment of inter-disciplinary organizational strategies and software engineering activities. Apart from its general and specific limitations, the organizational maturity model presented in this paper contributes significantly to the effectiveness of software product lines by addressing a topic of immense importance.

## References

- Ahmed, F., Capretz, L. F., & Sheikh, S. A. (2007). Institutionalization of software product line: An empirical investigation of organizational factors. *The Journal of Systems and Software*, 80(6), 836–849.
- Argyris, C. (1977). Double-loop learning in organizations. *Harvard Business Review*, 55, 115–125.
- Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., & Schmid, K., et al. (1999). *PuLSE: A methodology to develop software product lines*. In Proceedings of the 5th ACM SIGSOFT Symposium on Software Reusability, (pp. 122–131).
- Beckhard, R., & Harris, R. T. (1987). *Organizational transitions: Managing complex change*. Reading, MA: Addison-Wesley.
- Birk, G. H., John, I., Schmid, K., von der Massen, T., & Muller, K. (2003). Product line engineering, the state of the practice. *IEEE Software*, 20(6), 52–60.
- Bosch, J. (2000). *Design and use of software architectures: Adopting and evolving a product-line approach*. Reading, MA: Addison Wesley.
- Bosch, J. (2001). *Software product lines: Organizational alternatives*. In Proceedings of the 23rd International Conference on Software Engineering, (pp. 91–100).
- Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multi-method matrix. *Psychological Bulletin*, 56(2), 81–105.
- Cao, G., Clarke, S., & Lehane, B. (2000). A systematic view of organizational change and TQM. *The TQM Magazine*, 12(3), 186–193.
- Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1, 245–276.
- Clements, P. C. (2001). *On the importance of product line scope*. In Proceedings of the 4th International Workshop on Software Product Family Engineering, (pp. 69–77).
- Clements, P. C., Jones, L. G., Northrop, L. M., & McGregor, J. D. (2005). Project management in a software product line organization. *IEEE Software*, 22(5), 54–62.
- Clements, P. C., & Northrop, L. M. (2002). *Software product lines practices and patterns*. Reading, MA: Addison Wesley.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, 37–46.
- Comrey, A. L., & Lee, H. B. (1992). *A first course on factor analysis*. Hillsdale: Lawrence Erlbaum Associates.
- Crewson, P. (1997). Public service motivation: Building empirical evidence of incidence and effect. *Journal of Public Administration Research and Theory*, 7, 499–518.
- Cronbach, L. J. (1951). Coefficient alpha and the internal consistency of tests. *Psychometrika*, 16, 297–334.
- Dikel, D., Kane, D., Ornburn, S., Loftus, W., & Wilson, J. (1997). Applying software product-line architecture. *IEEE Computer*, 30(8), 49–55.
- El Emam, K. (1999). Benchmarking kappa: Inter-rater agreement in software process assessments. *Empirical Software Engineering*, 4(2), 113–133.
- Gordon, J. R. (2002). *Organizational behavior: A diagnostic approach*. New Jersey: Prentice Hall.
- Hames, R. D. (1994). *The management myth*. Sydney: Business and Professional Publishing.
- Hellriegel, D., Slocum, J. W., Jr., Woodman, R. W., & Bruning, N. S. (1998). *Organizational behavior*. Canada: ITP Nelson.
- Jacobsen, I., Griss, M., & Jonsson, P. (1997). *Software reuse—architecture process and organization for business success*. Reading, MA: Addison Wesley.
- Jazayeri, M., Ran, A., & van der Linden, F. (2000). *Software architecture for product families: Principles and practice*. Reading, MA: Addison Wesley.
- Jehn, K. A. (1995). A multi-method examination of the benefits and detriments of intra-group conflict. *Administrative Science Quarterly*, 40, 256–282.
- Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20, 141–151.
- Kaiser, H. F. (1970). A second generation little jiffy. *Psychometrika*, 35, 401–417.
- Kilmann, R. H., Saxton, M. J., & Serpa, R. (1985). *Gaining control of the corporate culture*. San Francisco, CA: Jossey-Bass.
- Koh, E., & Kim, S. (2004). *Issues on adopting software product line*. In Proceedings of the 11th Asia-Pacific Conference on Software Engineering, (pp. 589).
- Kottler, J. (1994). *Beyond blame: A new way of resolving conflicts in relationships*. San Francisco: Jossey-Bass.
- Kotter, J. P., & Heskett, J. L. (1992). *Corporate culture and performance*. New York, NY: The Free Press.

- Kuvaja, P. J., Simila, J., Krzanik, L., Bicego, A., Saukkonen, S., & Koch, G. (1994). *Software process assessment and improvement—the bootstrap approach*. Oxford: Blackwell.
- Landis, J., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33, 159–174.
- Lee, H. Y., Jung, H. W., Chung, C. S., Lee, J. M., Lee K. W., & Jeong, H. J. (2001). *Analysis of inter-rater agreement in ISO/IEC 15504-based software process assessment*. In Proceedings of the 2nd Asia-Pacific Conference on Quality Software, (pp. 341–348).
- Lyles, M. A. (1994). An analysis of discrimination skills as a process of organizational learning. *The Learning Organization*, 1(1), 23–32.
- Macala, R. R., Stuckey, L. D., Jr., & Gross, D. C. (1996). Managing domain-specific, product-line development. *IEEE Software*, 13(3), 57–67.
- Mannion, M. (2002). *Organizing for software product line engineering*. In Proceedings of the 10th International Workshop on Software Technology and Engineering Practice, (pp. 55–61).
- Marquardt, M., & Reynolds, A. (1994). *The global learning organization*. Burr Ridge, IL: Irwin.
- Mathieu, J. E., & Zajac, D. (1990). A review and meta-analysis of the antecedents, correlates, and consequences of organizational commitment. *Psychological Bulletin*, 108, 171–194.
- Medina, F. J., Munduate, L., Dorado, M. A., & Martínez, I. (2005). Types of intra-group conflict and affective reactions. *Journal of Managerial Psychology*, 20(3/4), 219–230.
- Nunnally, J. C., & Bernstein, I. A. (1994). *Psychometric theory*. New York: McGraw Hill.
- O'Reilly, C., & Chatman, J. (1996). Culture as social control: Corporation, cults, and commitment. *Research in Organizational Behavior*, 8, 157–200.
- Osterhof, A. (2001). *Classroom applications of educational measurement*. NJ: Prentice Hall.
- Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model version 1.1. *IEEE Software*, 10(4), 18–27.
- Rosen, R. (1995). *Strategic management: An introduction*. London, UK: Pitman.
- Schein, E. H. (1988). *Organizational psychology*. Englewood Cliffs, NJ: Prentice Hall.
- Stevens, J. (1986). *Applied multivariate statistics for the social sciences*. Hillsdale, NJ: Lawrence Erlbaum.
- Todd, A. (1999). Managing radical change. *Long Range Planning*, 32(2), 237–244.
- Toft, P., Coleman, D., & Ohta, J. (2000). *A cooperative model for cross-divisional product development for a software product line*. In Proceedings of the 1st International Conference on Software Product Lines, (pp. 111–132).
- van de Ven, A. H., & Ferry, D. L. (1980). *Measuring and assessing organizations*. New York: Wiley.
- van der Linden, F. (2002). Software product families in Europe: The Esaps & Café projects. *IEEE Software*, 19(4), 41–49.
- van der Linden, F., Bosch, J., Kamsties, E., Känsälä, K., & Obbink, H. (2004). *Software product family evaluation*. In Proceedings of the 3rd International Conference on Software Product Lines, (pp. 110–129).
- Verlage, M., & Kiesgen, T. (2005). *Five years of product line engineering in a small company*. In Proceedings of the 27th International Conference on Software Engineering, (pp. 534–543).
- von Eye, A., & Mun, E. Y. (2005). *Analyzing rater agreement manifest variable methods*. London: LEA Publishers.
- Walls, J. A., & Callister, R. R. (1995). Conflict and its management. *Journal of Management*, 21(3), 515–558.
- Wang, Y., & King, G. (2000). *Software engineering processes: Principles and application*. New York: CRC Press.
- Weiss, D. M., & Lai, C. T. R. (1999). *Software product line engineering: A family based software development process*. Reading, MA: Addison Wesley.
- White, D. H. (1982). *Contemporary perspectives in organizational behavior*. Boston: Allyn and Bacon, Inc.
- Wilson, A. M. (2001). Understanding organizational culture and the implication for corporate marketing. *European Journal of Marketing*, 35(3/4), 353–367.
- Wilson, D. C., & Rosenfeld, R. H. (1990). *Managing organizations*. London: McGraw-Hill.
- Witherspoon, P. D. (1997). *Communicating leadership—an organizational perspective*. Boston: Allyn and Bacon.

## Author Biographies



**Faheem Ahmed** received his M.S. (2004) and Ph.D. (2006) in Electrical Engineering from the University of Western Ontario, London, Canada. Currently, he is an assistant professor at College of Information Technology, UAE University, Al Ain, United Arab Emirates. Ahmed had many years of industrial experience holding various technical positions in software development organizations. During his professional career, he has been actively involved in the life cycle of software development process including requirements management, system analysis and design, software development, testing, delivery, and maintenance. Ahmed has authored and coauthored many peer-reviewed research articles in leading journals and conference proceedings in the area of software engineering. Ahmed's current research interests are Software Product Line, Software Process Modeling, Software Process Assessment, and Empirical Software Engineering. He is a member of IEEE.



**Luiz Fernando Capretz** has over 20 years of experience in the software engineering field as a practitioner, manager, and educator. Currently, he is an associate professor and director of software engineering program at Department of Electrical and Computer Engineering, University of Western Ontario, London, Canada. Before joining the University of Western Ontario, in Canada, he has worked since 1981 at both the technical and managerial levels, taught and carried out research on the engineering of software in Brazil, Argentina, England, and Japan. He was the Director of Informatics and Coordinator of the computer science program at two universities (UMC and COC) in the State of Sao Paulo/Brazil. He has authored and coauthored over 50 peer-reviewed research papers on software engineering in leading international journals and conference proceedings and has coauthored the book, *Object-Oriented Software: Design and Maintenance*, published by World Scientific. His current research interests are software engineering (SE), human factors in SE, software estimation, software product lines, and software engineering education. Dr. Capretz received his Ph.D. from the University of Newcastle upon Tyne (U.K.), M.Sc. from the National Institute for Space Research (INPE-Brazil), and B.Sc. from UNICAMP (Brazil). He is a senior member of IEEE.