Software Qual J (2010) 18:299–319
DOI 10.1007/s11219-009-9091-x

Knowledge-oriented software engineering process in a multi-cultural context

Hannu Jaakkola · Anneli Heimbürger · Petri Linna

Published online: 5 December 2009

Abstract In software engineering, leading trends can be detected that will affect the characteristic features of a product and its development process. On a product level, the growth of size and complexity is apparent—but on the one hand only. On the other hand, there is also a growing demand for simple and reasonable small software products executed by handheld terminals and smartphones; these applications are in many cases expected to collaborate with databases over the Internet. In addition, different kinds of service concepts (ASP, SaaS) are becoming recognized alternatives to the traditional way of buying software. Increasingly, software products are also distributed in a wide geographical scope to users with different cultural backgrounds and expectations. In software engineering work, as a consequence of this growth in size and complexity, the development work is more and more often distributed. The software business itself is becoming global because of acquisitions, offshoring, and international subcontracting. The globalization of work sets new requirements to the engineering processes: in international teams the organisational and cultural differences of the development subteams have to be recognized. In this paper, the focus is on the software development and its global dimension—especially the roles of multi-cultural and cross-organizational issues in software engineering. Our paper presents the results of the first phase of our three phases research project related to "Culture-Aware Software Engineering." The main result of the first phase is the multi-cultural software engineering working model introduced in our paper. Culture is seen as one example of the context, i.e. the situation at hand. The concept of culture has also different meanings, which have to be understood in well-organized software engineering. Software engineering work is analyzed as a knowledge creation process, in which both explicit and tacit

H. Jaakkola (✉) · P. Linna
Tampere University of Technology, Tampere, Finland
e-mail: hannu.jaakkola@tut.fi

P. Linna
e-mail: petri.linna@tut.fi

A. Heimbürger
University of Jyväskylä, Jyväskylä, Finland
e-mail: anneli.a.heimburger@jyu.fi

knowledge are recognized and the transformation between these establishes baselines along the development life cycle.

Keywords Software engineering process · Context · Culture · Knowledge

1 Introduction

This paper introduces a *three-layer model* as a framework to manage the complexity of software engineering work in multi-cultural organizations. As background, we will focus on clarifying the characteristics of Software Engineering, the concept of culture and the knowledge creation process. Section 1 (Introduction) introduces and specifies the key terminology used in our paper.

1.1 Software engineering

The term "Software Engineering (SE)" has several definitions. The IEEE and ACM (1990) define it as the

> "application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software."

The definition separates the activities included in the whole (development, operation, maintenance) and specifies quality properties (systematic, disciplined, quantifiable) for the work. From a technical point of view, the complexity of the work is typically managed by life cycle models, e.g. Pfleeger and Atlee 2006, pp. 49–58.

- The original *waterfall model*: specifies the SE Process as a flow of phases having a forward and backward (iteration) flow specified.
- *V-Model* is a modification of the waterfall model: it focuses on the bindings between the development and testing phases/baselines and also points out the importance of early test design closely connected to the development work.
- *Incremental/iterative (evolutionary) model*: emphasizes the construction of software either as increments, implementing new properties (i.e. incremental), or by expanding it with more detailed properties such as evolutionary stages in the development (iterative); in both cases the basic development flow is based on the waterfall type model.
- *Spiral model* (Boehm 1988): combines the elements of asset evolution and prototyping-in-stages, as well as risk management and the iterative approach to development cycles. This model will be discussed in more detail later (Sect. 4) in this paper.
- *RUP model* (Jacobson et al. 1999; Kruchten 1998): an iterative software development process framework created by the Rational Software Corporation to bind the SE process to utilize the UML modeling language in a proper way. In the RUP, the SE process is divided into four sequential development phases. Every phase consists of iterations—the application of six engineering disciplines. In addition, three supporting disciplines cover the whole development cycle of the software product.

In addition to the above, there are several modifications and applications of "basic life cycle models". The common feature they all share is the goal to increase understanding of the SE process and increase its manageability and that of the product.

An alternative—or in fact complementary—approach is the software engineering process model, in which the activities relevant to a *software organization* are recognized as individual *processes*. The aim of these process models is to provide a means for Software Process Improvement (SPI). The best known and most widely used process models are CMMI (SEI 2006) and SPICE (ISO/IEC 2004, 2006). SPICE (also called ISO 15504 according to the standard specifying its assessment principles) recognizes nine *process groups* divided into three *process categories*. In the assessment, each process is handled as a separate improvement item. CMMI has two basic approaches for process improvement—the *staged and the continuous* model. In both, similar types of processes to those in SPICE are recognized; in the staged model, the improvement path is "pre-specified", and the continuous model closely follows the same principles as SPICE.

There are two development approaches in the public discussion of modern software engineering, see e.g. (Boehm 2006):

- plan-driven development and
- agile development.

The *plan-driven approach* emphasizes the importance of following the pre-defined process, the importance of the ability to repeat the development path and the importance of documentation for tracing decisions. *Agile development* is based on the principles of the "Agile Manifesto" (Agile 2009). The manifesto was originally announced in 2001 and it has its implementation in various new methodologies such as Extreme Programming (XP 2009) and SCRUM (Scrum 2009). Agile methodologies generally promote a project management process encouraging teamwork and peer-reviews. It is also based on a new type of leadership culture, which takes into account continuous communication, visibility of the project progress, workload of the project staff, etc. The work is planned in short iterations and continuous integration of software assets in the whole, including integration testing. The goal is to deliver high-quality software based on the approach that aligns development in close collaboration with customers and rapid reaction to changes in the requirements.

In his ICSE conference paper (2006), Boehm provides an excellent overview of the current trends in software engineering. These include the following:

- The use of technologies providing a higher abstraction level to manage the complexity of software (Object technologies, Modularity and manageable interfacing between modules, Service-Oriented Architectures, etc.).
- The increasing importance of human factors (user value) in the development of information systems (Value-Based SE).
- Along with the plan-driven development culture, agile methods are widely adopted by the software industry.
- The growth of software complexity and criticality. Boehm uses the term "Complex Systems of Systems" to point out the importance of understanding the software as a part of the integrated whole and implemented in a wide variety of platforms.
- Globalization of work—distributed organizations and cultural diversity in the development work.
- Systematic (organized) reuse is needed to provide competitive, maintainable, and high-quality software products.

As a consequence, there is a need for *increased communication* and the organization of work as a collaborative activity of distributed teams having members of diverse cultural backgrounds.

1.2 Culture

Terminology within the multicultural and cross-cultural communications field seems sometimes to be inconsistent when reading the literature, websites, or other materials. One may ask, what is the difference between intercultural and cross-cultural? What is cross-cultural awareness as opposed to cross-cultural knowledge, or are cultural sensitivity and cultural competence the same thing (Kwintessential 2009; Intercultural Communication 2009)? The key concepts are summarized in Table 1.

As seen in Table 1, the key concepts in this context are numerous. The term "culture-aware" is used as an umbrella term to include all aspects relevant to the cross- and multicultural contexts.

1.3 Structure of the paper

In this paper, we will concentrate on the software engineering process in a multicultural context. Software engineering is seen as a *knowledge-creation process*, in which the communication and transfer of knowledge from one interest group to another plays a central role. The difficulty of software engineering increases while the level of distribution grows. Partially, the problems of distribution of work are solved by tools (e.g. version management), the right kind of division of work (independency of modules), and a management culture that takes into account the distributed character of the development team. An additional difficulty is caused by the multicultural character of the team. People of different cultural backgrounds expect different kinds of leadership and ways of communication. In many cases, their ability to communicate with each other is limited because they lack a common language; therefore, the communication assets used must support multicultural communication.

Our paper presents the results of the first phase of our three-phase research project related to Culture-Aware Software Engineering. The main result of the first phase is the multicultural software engineering working model introduced in our paper. Our model integrates several wide research domains: software engineering processes, knowledge management, context, and culture research. For this reason, we think that it is also worth describing the background of our model. In the next research phases, we will focus on implementing and testing our model as well as on iterative model development based on the results of testing.

In Sect. 2 of our paper, we will provide a review of the related studies in the area of cultural differences and cross-cultural communication. Special focus is given to the reported experiences in multicultural software engineering work. In Sect. 3, we will specify our knowledge-creation model, which is based on the studies of Nonaka and Takeuchi (1995). We will also discuss the concept of context and different multicultural contexts. Section 4 introduces our Multicultural Software Engineering Working Model. Section 5 concludes the paper.

2 What is culture: related work

Culture is a problematic concept. It has been studied for hundreds of years, but there is still no clear definition for it. The word culture has been used inexactly, and different scholars (i.e. anthropology, pedagogic, psychology, and sociology) have approached culture from different directions. This is one of the reasons why there are so many definitions.

Table 1 Key concepts

Concept	Description
Culture	According to Hofstede (2003); Hofstede and Hofstede (2004), culture is a collective phenomenon, because it is shared with people who live or lived within the same social environment, which is where it was learned. Culture consists of the unwritten rules of the social game. It is the collective programming of the mind that separates the member of one group or category of people from others. According to King (2007), cultures can be considered at four levels: national cultures, organizational cultures, organizational subcultures, and subunit cultures. We extend the concept of culture by domain cultures, project cultures, team cultures, and task cultures
Cross-cultural	Cross-cultural describes comparative knowledge and studies of a limited number of cultures (Lewis 1999)
Multicultural	Multicultural describes comparative knowledge and studies of relating to, or including several cultures (Lewis 1999)
Intercultural communication	This is in many ways similar to group communication, but the role of groups is taken by ethnic cultures. Culture, of course, is not just the domain of nations; it also describes the norms and conventions of groups (such as team culture), and collectivities with shared knowledge and ideology (such as academic culture). However, as it is used in communication studies, intercultural communication tends to describe the relations between members of different ethnic groups and languages, interacting in an international context, such as in international software project teams. (Marsen 2006)
Cross-cultural knowledge	Cross-cultural knowledge is critical to basic cross-cultural understanding. Without it cross-cultural appreciation cannot take place. It refers to a surface level familiarization with cultural characteristics, values, beliefs, and behaviors. (Kwintessential 2009)
Cross-cultural knowledge space	Cross-cultural knowledge spaces are personal and collaborative virtual working environments on fixed or ubiquitous Web and in physical worlds like in meeting rooms, especially designed for cross-cultural teams. The space can include applications such as knowledge management services, team calendars, project monitoring functions and electronic cultural assistants running in personal and collaborative information systems. (Heimbürger 2008a, b)
Cross-cultural understanding	Cross-cultural understanding simply refers to the basic ability of people within projects and teams to recognize, interpret, and correctly react to people, incidences or situations that are open to misunderstanding due to cultural differences. (Kwintessential 2009)
Cross-cultural awareness	Cross-cultural awareness develops from cross-cultural knowledge as the learner understands and appreciates a culture internally. This may also be accompanied by changes within the learner's behavior and attitudes such as a greater flexibility and openness. (Kwintessential 2009)
Cross-cultural sensitivity	Cross-cultural sensitivity is a natural by-product of awareness and refers to an ability to read situations, contexts, and behaviors that are culturally rooted and be able to react to them appropriately. A suitable response necessitates that the actor no longer carries his/her own culturally determined interpretations of the situation or behavior (i.e. good/bad, right/wrong), which can only be nurtured through both cross-cultural knowledge and awareness. (Kwintessential 2009)

Table 1 continued

Concept	Description
Cultural competence	Cultural competence is a developmental process that evolves step-by-step over an extended period. Both individuals and organizations are at various levels of awareness, knowledge, and skills on the cultural competence continuum and levels. Cultural competence is about respecting cultural differences and adapting to changing situations and benefiting from them. Competence is the final stage of cross-cultural understanding and signifies the actor's ability to work effectively across cultures. Cross-cultural competency is beyond knowledge, awareness and sensitivity in that it is the digestion, integration, and transformation of all the skills and information acquired through them, applied to create cultural synergy within the workplace. (Kwintessential 2009)
Context	Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. (Dey et al. 2005)
Distributed software engineering	Distributed software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software at multiple sites. (Jaakkola and Thalheim 2005)

Anthropology has the longest history of studying culture, and the first theories about culture were made by its sub-branch of ethnology. Ethnology is almost synonymous with cultural anthropology in Europe, as it studies cultural variation among humans. As a result of the long history of culture research, Kroeber and Kluckhohn (1952) collected culture definitions and found in 1952 that there were 164 cultural definitions. How many definitions must there be now? Research into culture has been difficult to conduct and hard to develop and refine (Straub et al., 2002). In many research papers, authors have tried to evaluate the meaning of culture. Hofstede (1984, pp. 21) defines culture as "The collective programming of the mind that distinguishes the members of one human group from another." Since there are so many definitions and theories, it is actually rather simple to use the word culture; it "includes" everything you want!

Culture consists of many elements i.e. artifacts, stories, heroes, symbols, values, beliefs, behavior, and underlying assumptions. Some of these attributes are visible, but some are tacit, which are deeper in the human mind. Attributes are more difficult to change awareness of and explain, depending how deep they are in the human psyche. Culture can be divided into cultural spheres such as industry, national, regional, organizational, professional, corporate, and functional. These spheres include different dimensions, i.e. the national cultures sphere includes five dimensions according to Hofstede and Hofstede (2004): individualism/collectivism, power distance, masculinity/femininity, uncertainty avoidance, long-term/short-term orientation. Hofstede's cultural pattern is used in many research papers, and his dimensions are sometimes also used for other than national cultural spheres. Putting all those cultural spheres together into a melting pot, results in the concept of the "virtual onion," in which different layers (spheres) change places depending on the context. Straub et al. (2002) suggest that the virtual onion is one's overall culture and its combination is unique for each individual. Personality is in the middle of this virtual onion, which also means that people rarely act in the same way in the same situations.

In many papers, researchers have tried to find the best common culture dimensions or the best culture dimensions for spheres. Marcus and Baumgartner (2004) list the five best dimensions: contexts, technology, uncertainty avoidance, time perception, and authority conception. Kluckhohn and Strodtbeck (1961) state that the best five dimensions are: relationship with time, human activity, human nature, relationships with people, and time. Which dimensions are the best for software quality management or software quality process improvement? As companies have started outsourcing processes and performing global collaborative teamwork, awareness of these dimensions has become more important (Krishna et al. 2004). Before deciding what those "best dimensions" are, it is good to know which aspects can cause conflicts, i.e. in cross-cultural communication in collaborative teamwork. Shah (2004) identifies six different factors (stumbling blocks): the assumption of similarities, language differences, non-verbal misinterpretations, preconceptions and stereotypes, the tendency to evaluate, and high anxiety. When communicating cross-culturally, we usually try to find more non-verbal cues, especially if using a non-native language (LeBaron 2003). Effective virtual teamwork would also need the team workers' willingness to communicate (WTC) with a different culture and to respect those cultural differences (Walker 2005).

Straub et al. (2002) classify definitions into three main groups: definitions based on shared values, definitions based on problem solving, and general all-encompassing definitions. In the study by Straub et al., the very commonly used definition of Hofstede's (2004) national cultures is categorized in the shared values category. Definitions can also be divided into objective and subjective categories (Hoft 1996). Different kinds of categories can be used to help understand several theories and their dimensions. From a software quality point of view, the cross-cultural dimensions have been integrated in software quality improvement by Siakas et al. (1999).

Tools, methods, and technologies are used to lower cultural barriers. Using models in communication, instead of oral or written communication, is one example of this. Because multicultural organizations are also in most cases distributed, the tools supporting time- and place-independent communication and collaboration are important. There are hundreds or even thousands of collaborative tools (Good 2009)—some of them also taking into account cultural differences. An interesting collaboration environment is the 3D-virtual environment that can be used to replace traditional meetings. These tools have their roots in virtual (game) communities—e.g. Habbo Hotel and Second Life, but are also used in an increasing amount in serious business and learning applications.

3 Knowledge creation and context

3.1 Knowledge creation

The term "tacit knowledge" was used for the first time in 1966 by the philosopher Michael Polanyi (1967). The term was made more well-known by Nonaka and Takeuchi (1995) in their book "The Knowledge-Creating Company." They introduced the SECI model to describe the creation process of new knowledge in an organization. It is based on the interaction of explicit and tacit knowledge in an organization to create new knowledge and innovations.

The main elements of the model are (Nonaka and Takeuchi 1995; Landow 2009; de Geyetere 2009):

Table 2 Tacit and explicit knowledge

	Tacit	Explicit
Tacit	Socialization	Externalization
	Ba: Originating	Ba: Interacting/dialoging
	KA: Sympathized/experimental	KA: Conceptual
Explicit	Internalization	Combination
	Ba: Exercising	Ba: Cyber/systemizing
	KA: Operational/routine	KA: Systemic

- SECI process;
- Ba's (knowledge-sharing contexts);
- Knowledge Assets (KA—resources creating value for the company).

The synthesis of the main elements is described in Table 2. The knowledge in an organization is classified as explicit and tacit. *Explicit knowledge* is the knowledge that can be recognized in an organization and is easy to communicate. *Tacit knowledge* is the "property" of individuals having a technical and cognitive dimension—the personal skills and know-how and the cognitive dimension—beliefs, ideals, values, schemata, and mental models. In the organization, the process of getting tacit knowledge to benefit the organization is the source of new innovations.

The *SECI model* describes the Knowledge Creation Spiral—the transformations between tacit and explicit knowledge in an organization. The spiral starts from the upper left part (Socialization), continues clockwise from one quadrant to the next (Externalization—Combination—Internalization) and after reaching the upper left quadrant starts a new cycle of knowledge creation.

Socialization (tacit to tacit) is transferring tacit knowledge in the form of social communication and interaction between individuals without language. It is based on informal communication and learning by example. Socialization takes place between people in meetings or in team discussions. *Externalization* (tacit to explicit) is developing concepts that allow communication of tacit knowledge in explicit form through dialog (e.g., brainstorming). Externalization is a process in which knowledge moves out from a person in the form of metaphors, analogies, concepts, hypotheses, and models. In a group, the individuals are committed to the group; the intentions and the ideas fuse and become integrated with the group's mental world—see *group culture* later in this paper. *Combination* (explicit to explicit) collects the explicit information in integrated concepts and systems. It covers the exchange of information in different pre-defined forms and channels, reconfiguration, manipulating, and reorganizing. Knowledge creation carried out in formal education and training usually takes this form. This transformation phase can be best supported by technology. Explicit knowledge can be easily captured and then distributed/transmitted to a worldwide audience. *Internalization* (explicit to tacit) is closely linked to learning by doing—the explicit knowledge becomes a part of an individual's mental model. It requires the individual to identify the knowledge relevant within one's self within the organizational knowledge. Learning by doing, training, and exercising allow the individual to access the knowledge of the group and the entire organization—see "organizational culture" later in this paper.

One significant goal of knowledge management is to create the technology to help users derive tacit knowledge from explicit knowledge. In the SECI spiral, the transformation repeats the creation cycles. The knowledge internalized by an individual in the form of know-how and skills (tacit knowledge) will be socialized by communication etc.

The *Knowledge assets (KA)* are company-based resources that are important to create values for the firm. They are inputs, outputs, and moderating factors of the knowledge creating process. A company has to "map" its inventory of knowledge assets and to be able to create new assets from old ones. The *Sympathized/Experimental KAs* are tacit knowledge assets acquired through common experiences (skills, know-how, etc.). *Conceptual KAs* are explicit knowledge assets articulated through images, symbols, and language. *Systemic KAs* are systemized and packaged explicit knowledge assets (documents, manuals, databases, patents, licenses). *Operative/Routine KAs* are tacit knowledge assets routinized and embedded in actions and practices (know-how in daily routines, organizational routines, and culture).

The concept of *Ba* is difficult to translate. The word is Japanese, and it has roots in Japanese philosophy. The term Ba was applied by Nonaka and Takeuchi to define a shared context, in which knowledge is shared, created, and utilized through interaction. The context space may be physical (office, project room), virtual (email, teleconference, joint workspace), mental (shared experience, ideas, ideals), or any combination of them. In general, it is a question of the "tools" supporting knowledge creation and manipulation of knowledge along the process. Every quadrant of the SECI process has a specific Ba (Table 2). *Originating Ba* is the world where individuals share feelings, emotions, experiences, and a mental model. Physical contacts are important in Originating Ba—e.g. in the form of meetings. *Interacting/Dialoging Ba* supports the transformation of an individual's mental models and skills into common terms and concepts. Initiators are challenged to pursue their ideas and it provides a platform for dialog, where people engage jointly in the creation of meaning and value. An example of this kind of context is brainstorming. *Cyber/Systemizing Ba* is a place for interacting in a virtual world instead of real space and time. Combining new explicit knowledge with the existing generates and systemizes explicit knowledge throughout the organization. It is supported by collaborative environments utilizing information technology—groupware, and Internet-based knowledge, etc. *Exercising Ba* is focused on training with senior mentors and expects the active participation of individuals in continuous self-refinement.

The knowledge-creation structure (including all its components and operating principles) was originally developed to understand the knowledge creation process of an organization in an innovative and development-oriented context. This model is used to analyze and structurize the complexity of the software engineering process, which typically includes components of new knowledge creation, knowledge mining and modeling, transfer of the knowledge from one interest group to other, etc. The components of the SECI model are used to specify and recognize the needs of different types of communication—knowledge exchange—in the heterogeneous community of the interest groups related to the different phases of the software development process. The spiral can also be used to describe the amount of new knowledge (growing area of the spiral), the resources used to create the knowledge, and the characteristics of the activities/support needed to mine the knowledge from the sources relevant to the quadrant. This view is very similar to the interpretation of the software process model discussed in Sect. 4 of this paper.

3.2 Knowledge has context

Humans can quite successfully express their thoughts and ideas to each other and react appropriately to them. There are several factors that have an effect on this, such as the versatility of the semantics in the language people use, a common culture, and common understanding of how the world works, as well as a tacit understanding of everyday situations.

Situational knowledge is knowledge that is specific to a particular occasion (Brézillion 2003). Some methods of generating knowledge, such as trial and error, or learning from experiences, tend to create highly situational knowledge. *Situational knowledge* is often embedded in language, culture, working methods or traditions. Humans are able to use implicit situational knowledge, when they interact with each other. By means of implicit situational knowledge, humans can increase their conversational dimensions. Situational knowledge is also known as context (Brézillion 2003).

Since we are social beings, one of our basic needs is to communicate with other humans. Communication usually happens within a context. A communicated piece of knowledge always has some relevance to another piece of knowledge or to a specific situation. Associations express the relation of one knowledge unit to others and thus express the context in which this knowledge unit is relevant. Without context, it is difficult to absorb a piece of knowledge and as a consequence, it probably will not reach the status of new knowledge.

There are several definitions of *context* (Ahn et al. 2005; Coutaz et al. 2005; Dey et al. 2005; Kiyoki et al. 1994, 2009). Many definitions of context are made by listing examples or by choosing synonyms for context. The notion of context is a fundamental concern in cognitive psychology, linguistics, and computer science. Our focus is on *computer science*. The computer science community has initially perceived context as a matter of the user's locations. In the last few years, this notion has been considered not simply as a state but also as a part of a process in which users are involved. In order to use context effectively, we must understand what context is and how it can be used. Existing research on context can be classified into two main categories: (a) context-based *delivery* of knowledge and (b) the *capture and utilization* of contextual knowledge. An understanding of context will enable application designers to choose what context to use in their application.

Almost all information available at the time of an interaction can be seen as contextual information. Some examples are identity, task at hand, spatial, temporal, and environmental information, social situation, nearby resources, physiological measurements, feelings, and impressions. Brézillion (2003) has introduced the concepts of contextual knowledge and proceduralized contexts. Contextual knowledge is background knowledge. Proceduralized context is immediately useful for the task at hand, for example, a person can analyze the situation, collect contextual information, identify the context in which the incident is occurring, and decode what to do. An interesting issue is the transitional stage from contextual knowledge to proceduralized context. This transition is *context's dynamic dimension*. Links between knowledge and context can be described as follows: (a) context is knowledge and knowledge is context, (b) context is defined and structured with respect to a focus of attention, (c) context granularity depends on the distance to the focus of attention, (d) context structure evolves dynamically with the evolution of the focus of attention, and (e) context is relative to an observer.

Context-oriented knowledge management provides a structured approach for knowledge transformation, by means of (a) the knowledge context that models the characteristics of the explicit knowledge, (b) the conceptual context that models the structure of the implicit knowledge, and (c) the physical context that determines the physical environment in which the knowledge transfer will take place (Brézillion 2003; Nonaka and Teece 2001). Context objects must be modeled and defined as specific entities at some point, but this decision should be deferred to the design of concrete applications or perhaps even to run-time where users should be free to specify the nature and format of context objects.

From the viewpoint of software engineering projects, processes and teams, in the era of the World Wide Web (WWW), a large number of *knowledge resources* are distributed in

the worldwide network environment. In this environment, one of the most important issues is how the user can extract appropriate knowledge according to his/her context. As discussed, the concept of context is diverse. We focus here on an operational definition of context and its applications to intellectual teamwork—new knowledge creation and sharing among members in a cross-cultural environment. In our approach, context is knowledge and knowledge has context. Context is all about the whole situation relevant to an application and its set of users. Context can emerge in the moment and it can change quickly (Heimbürger 2008b; Heimbürger 2007). Contexts can be static, dynamic, discrete, continuous, individual, or shared (as Ba in Nonaka and Takeuchi SECI model). Context is any information and/or knowledge that can be used to characterize the situation of an entity. An entity is an artifact that is considered relevant to the interaction between a user, a process and an application.

3.3 Multicultural context

Globalization is one of the main trends in our world. Software engineering is a good example of the intellectual teamwork that is nowadays more and more geographically distributed, and cross-cultural because of globalization. Increasingly, eastern and western cultures meet each other in connection with software engineering business projects. Cross-cultural actions are carried out both in the virtual world and in the physical world e.g. via e-mails, Web meetings, and collaborative virtual working spaces as well as in face-to-face meetings. In situations like these, we may experience some cultural differences relating to business and social etiquette, meeting protocols, formality and rituals, orientation to time, communication style, working methods, and the decision-making process.

Cultural sensitivity has become an important dimension for success in today's international software business and research arena. Despite the trend of globalization, software business executives, project managers and project team members are finding themselves in uncertain situations due to culturally dependent differences in communication protocol, language, and value systems. Consequently, people involved in cross-cultural transactions are advised to be aware of the cultural backgrounds of their counterparts. Cultural competence might help for example software project managers to achieve project goals and avoid potential risks in cross-cultural software project environments and would also support them to promote creativity and motivation through flexible leadership. By understanding some of the main cultural dimensions and by adjusting to cultural differences, software engineers can face the challenge and become better negotiators, project managers, and team members on behalf of their companies and research organizations.

When we are talking about the concept of culture, it is very important to understand its different levels. According to King (2005), cultures can be considered at four levels: national cultures, organizational cultures, organizational subcultures, and sub-unit cultures. We have extended King's categorization with team cultures.

Organizational culture is characterized by consistency across individuals and units in terms of assumptions, values, and artifacts. *Assumptions* are formed over time as members of an organization make decisions, cope with problems, and take advantage of opportunities. *Values* are a set of social norms. *Artifacts* are the visible aspects of an organizational culture, for example a knowledge repository system. Organizational subcultures may reflect organizational structure, professional occupations, task assignments, rank in hierarchy, or technologies used. Sub-unit cultures are created within the boundaries of particular subunits of an organization. Team cultures are mechanisms for individuals with diverse specialized knowledge to work toward a common goal. Teams are typically

focused on a single objective, and they are temporary. If all team members are from the same organizations, the team culture reflects the organizational culture. In multi-organizational projects, many team cultures may collide or softly meet depending on the cultural competence of the team members and the ICT systems they are using.

How do cultures relate to knowledge management? Holden (2002) discusses how culture shapes assumptions about which knowledge is important. Culture mediates the relationships between organizational and individual knowledge. *Culture creates a context* for social interaction. Culture shapes processes for the creation and adoption of new knowledge. Integrating cross-cultural knowledge into SE projects and processes is a challenge when developing models, methods and practices for the new, cross-cultural software engineering field.

In Sect. 4, we will integrate our study about knowledge, context and multiculturality by introducing an integrated Multicultural Software Engineering Working Model.

4 Multicultural software engineering (MCSE) working model

Our Multicultural Software Engineering Working Model has three *interacting* main layers: (a) a software engineering process layer (based on the spiral model), (b) a knowledge-context layer, and (c) a multi-cultural context layer. We describe the layers as follows and also give a hypothetical example of implementing our model.

4.1 The layers of the multicultural software engineering working model

4.1.1 Software engineering process layer

In Sect. 1 (Introduction), some basic life cycle and process models for software engineering were discussed. The software engineering process layer is based on Boehm's (1988) *Spiral model*. The evolution spiral bears analogy with the SECI as a knowledge creation process. The analogical features are: every spiral cycle increases knowledge and understanding of the product developed; the size of the spiral can be interpreted as the usage of resources; every quadrant in the spiral cycle creates "specialized" knowledge of the product or process. In this approach, we do not clearly separate the KAs and Bas supporting the knowledge; these are available in the next layer of the model.

In spite of its early date of publication, Boehm's (1988) spiral model includes essential elements and a message for the modern software engineering work. It can be applied as it is, but its principles may be embedded in any development process model applied by the organization. The motivation of Boehm to introduce the spiral model (Fig. 1) was his criticism of the traditional (waterfall model based) way of managing software projects. The main reasons were related to its inability to react to schedule delays and in the realization of project/product related risks. The use of prototypes made the development process iterative and built-in risk analysis provided the means to react to problems encountered early enough.

In the Spiral model, the development phases are represented in cycles starting from the inner part of the spiral. Every cycle starts by determining the objectives, alternatives, and constraints of the product being elaborated (upper left quadrant). These provide a means for the most important part of the model—risk analysis. It is *creating risk knowledge* and allows continuation of the project only if the risks encountered are at an acceptable level and manageable; otherwise the project is halted. Prototyping, simulation, and benchmarking are used to realize the properties of the baseline product and features of the

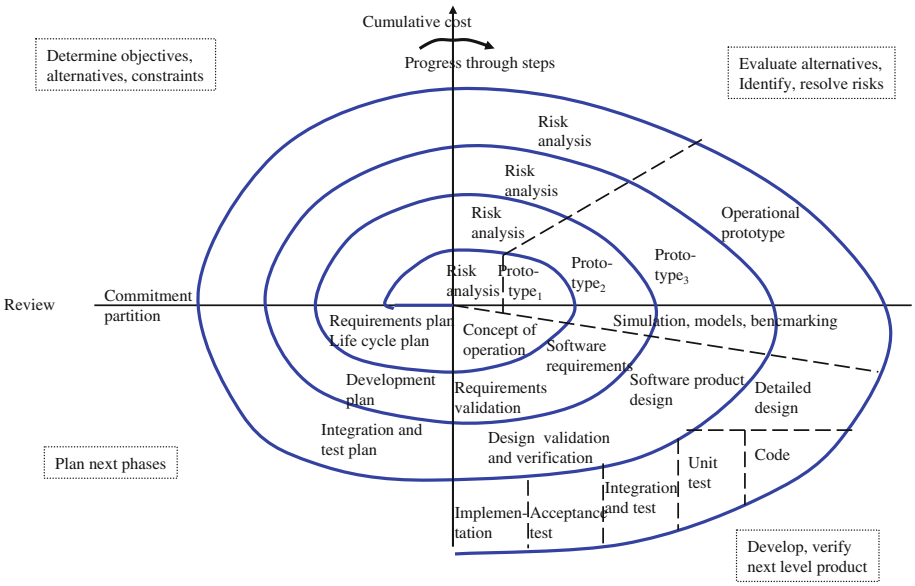

Fig. 1 Spiral model of software process (Boehm 1988)

development process related to it. Every cycle represents a development phase (requirements, design, and implementation). The evolution of the product is based on prototyping, simulations, and benchmarking; acceptance of every phase (cycle) is based on careful verification and validation. The activities of the development cycles are included in the lower right quadrant of the model; verification and validation of the baseline product comes at the end of this part. The lower left quadrant ends the cycle and concentrates on planning the next phases. The cycle ends in a review and commitment to the decisions (implemented as baseline products) made during the cycle.

4.1.2 Knowledge-context layer

The second layer of our model is the knowledge layer. Our knowledge-context layer includes the SECI model that describes the Knowledge Creation Spiral. We have extended the SECI model with contextual knowledge (Table 3). Each knowledge transformation step includes examples of typical KAs and Bas relevant to it.

4.1.3 Multicultural context layer

The third layer of our model is the multicultural context layer. In this layer, we have identified five temporal levels that describe the need for multicultural knowledge i.e. multicultural service as a function of time (Table 4).

The multicultural service differs on each level. On the Alarm and Short-term levels, the service is specially designed and implemented for urgent needs (Heimbürger et al. 2009; 2009a). For the Middle and Long-term levels, the user can also consult the websites of Hofstede and the Lewis cultural models. On the Eternity level, the user also has time to study books and articles. Multicultural services can also be designed for the continuum novice to expert in multicultural competence.

Table 3 Integrated spiral knowledge-context layer

Socialization: *tacit to tacit*	**Externalization**: *tacit to explicit*
Context-sensitive	*Context-sensitive*
Contextual knowledge is related for example to national, organizational, project and team cultures	Context knowledge is for example reported knowledge such as working protocols in an SE company
Multicultural SE project team working and communication:	*Multicultural SE project team working and communication:*
Ba (originating): face-to-face informal discussions	Ba (interacting/dialoging): one-to-many or one-to-one, formal communication
KA (sympathized/experimental): non-verbal communication	KA (conceptual): verbal communication, explicit knowledge articulated through symbols
Internationalization: *explicit to tacit*	**Combination**: *explicit to explicit*
Context-sensitive	*Context-free*
Context knowledge is for example navigational knowledge in complex systems based on specifications	Context knowledge: none (examples are formal languages, mathematics, physics etc.)
Multicultural SE project team working and communication:	*Multicultural SE project team working and communication:*
Ba (exercising): informal and/or formal, reading SE documentation and extending the ideas	Ba (cyber/systemizing): ICT based formal communication
KA (operational/routine): practice-based communication, cultural issues embedded in practices and routines	KA (systemic): formal language for information/knowledge access

Table 4 Multicultural services as a function of time

Temporal levels	Multicultural service
"Alarm" level—help me NOW!	Our e-assistant for cross-cultural communication
Short term level—situation is coming up soon (within some hours)	Our e-assistant for cross-cultural communication
Middle term level—the situation is coming up within one or 2 weeks	Our e-assistant for cross-cultural communication, the website of Lewis' model of culture, Hofstede website
Long term level—the situation is coming up in the near future (within 2–3 months)	Our e-assistant for cross-cultural communication, the website of Lewis' model of culture, Hofstede website, books and articles
Eternity level—the situation may come up within the next few years	Books, articles and websites

4.1.4 The three-layer model

In our Multicultural Software Engineering Working Model, the three layers interact with each other. The order of the layers from bottom-up is: (1) the spiral model of the software process, (2) the integrated spiral knowledge-context layer, and (3) the multicultural service layer.

In Fig. 2, the arrows from top-to-bottom describe the SE tasks to be carried out starting from the multicultural layer. The arrows from bottom-to-top describe multicultural practices that have been developed during the task realization among the SE team. This cultural tacit knowledge will be stored in the multicultural service layer.

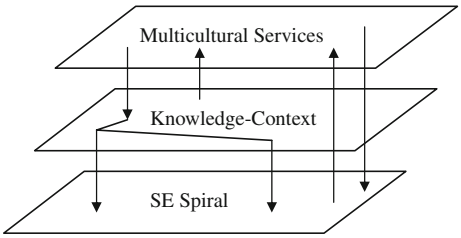

Fig. 2 The three layers of the multicultural SE working model

4.2 A hypothetical example of implementing the multicultural software engineering working model

Let us study the following example (clarified by the interaction described in Fig. 2). An SE project team member is going to work in a multicultural team. The task of the team is to write a software requirement plan. In the spiral model layer, the task is in the core of the spiral. In the knowledge-context layer, the situation is both in the block Socialization: Tacit to Tacit and in the block Externalization: Tacit to Explicit. The important point is that the team members first have to pass the multicultural service layer before they can proceed to other layers in their task. When logging into the system, they can define their cultural competence level and also the temporal level of the need for the multicultural service. They can browse cultural information or study cultural knowledge in more depth. They can also create a tacit knowledge base of cultural practices that are based on their own experiences. This can be done at a team level as well. For example, the system can give the team member the following information related to Finnish, Czech, and Japanese ways of communication, meeting and team working behavior (Bijl 1995; de Mente 2001; Gorrill 2008; March 1990).

4.2.1 Finland

Communication: Finns are transactional and do not need long-standing personal relationships in order to conduct business. The basic business style is formal; the Finns prefer people to speak succinctly and to focus purely on business. Finns do not require face-to-face contact and, in fact, are quite comfortable using e-mail. Finns are excellent time managers who prefer to organize their workday in order to accomplish as much as possible. Finns are interested in long-term relationships. Relationship building often takes place outside the office: in a restaurant or sauna. Never turn down an invitation to use the sauna, as it is an entrenched part of the Finnish culture. Finns place a great value on speaking plainly and openly. What someone says is accepted at face value and this is a culture where "a man's word is his bond" and will be treated as seriously as a written contract, so verbal commitments are considered agreements. Finns are direct communicators. Expect your colleagues to tell you what they think rather than what you want to hear. Professional differences are not viewed as personal attacks.

Meetings: Appointments are necessary and should be made in advance by telephone, e-mail, or fax. It is extremely difficult to meet people without a formal appointment. Meetings should not be scheduled between June and August as many Finns are on vacation during the summer. Arriving at meetings on time or slightly early is very important. A telephone call is expected immediately if the delay is more than 5 minutes. Punctuality is a

sign of respect and efficiency. There is very little small talk, if any, before getting into the business discussion. An agenda should be sent before the meeting as well as the biographies of the team. Meetings begin and end on time. Exaggerated claims should be avoided in presentations. Finns seldom ask questions. The presenter is expected to make his/her case with sufficient detail so that their Finnish colleagues do not need to ask questions. There is no taboo on humor in the business environment.

4.2.2 The Czech Republic

Communication: It will take several meetings for your Czech business associates to become familiar with you and appear comfortable and friendly. Business is conducted slowly. It is good to be patient and not appear ruffled by the strict adherence to protocol. Business is hierarchical. Decision-making power is held at the top of the company. Decisions are reached slowly. It may take several visits to reach a decision. High-pressure tactics should be avoided. Czechs generally offer what they expect to get and do not often give counter-offers.

Meetings: Appointments are mandatory and should be made in advance. Letters should be addressed to the company rather than a specific person. This prevents a letter from being held up if the person it is addressed to is away from the office. Meetings should not be scheduled on Friday afternoon as many Czechs leave for their country cottages after lunch. Punctuality for meetings is taken extremely seriously. Initial meetings are scheduled to get to know each other and to see if your Czech associates believe that you are trustworthy. The first meeting may be with a gatekeeper rather than the actual decision maker. Direct eye contact should be maintained while speaking. The Czechs do not remove their suit jacket unless the highest-ranking person does so. Presentations should be accurate, detailed and thorough, with charts and figures to back up particular claims. Businessmen wear suits. English is widely spoken, particularly among the younger generation, but it is a good idea to check if an interpreter will be needed in advance. Older people tend to speak German. Long business lunches are usual. Visits during July and August are worth avoiding as many businesses close for summer vacation, in particular in small cities.

4.2.3 Japan

Communication: The Japanese are non-confrontational. They have a difficult time saying 'no', so it is good to be vigilant at observing their non-verbal communication. It is best to phrase questions so that they can answer yes. For example, "do you disagree with this?" Group decision-making and consensus are important. Written contracts are required. The Japanese often remain silent for long periods of time, so patience is needed. Japanese prefer broad agreements and mutual understanding so that when problems arise they can be handled flexibly. Never lose your temper or raise your voice during negotiations. Some Japanese close their eyes when they want to listen intently. The Japanese seldom grant concessions. They expect both parties to come to the table with their best offer. The Japanese do not see contracts as final agreements so they can be renegotiated.

Meetings: The Japanese prefer to do business on the basis of personal relationships. In general, being introduced or recommended by someone who already has a good relationship with the company is extremely helpful as it allows the Japanese to know how to place you in a hierarchy relative to themselves. Appointments are required and, whenever possible, should be made several weeks in advance. It is best to telephone for an appointment rather than send a letter, fax, or email. Punctuality is important. Arrive on

Table 5 Implications of cross-cultural differences for team working

Hostede's dimensions	Implication
Individualism/ collectivism	Team members from a collectivistic society are likely to spend more time on long-term goals, are more likely to make realistic plans, and are more likely to be cooperative. Conversely, team members from individualistic societies are more likely to focus on the short-term, make extreme plans, are more likely to view projects from a fixed perspective, and are more likely to be competitive
Power distance	Team members from low power distance cultures may be frustrated by the need of team members from high power distance cultures to seek approvals from higher authority. On the other hand, team members from high power distance cultures may feel pressured by the pace imposed by team members from low power distance cultures
Masculinity/femininity	When collaborating, team members from masculine cultures are more likely to be competitive (win–lose) and those from feminine cultures to be empathic and seek compromise (win–win). This means that team members from masculine cultures are likely to view the feminine team members as "avoiding' while the feminine team members is likely to view their masculine negotiator as "contending."
Uncertainty avoidance	Team members from high risk avoidance cultures are likely to view those from low risk avoidance cultures as unfocused. Those from low risk avoidance cultures are likely to view team members from high risk avoidance cultures as rigid
Long-term/short-term orientation	Team members in long-term oriented cultures are accustomed to working toward building strong positions in their markets and do not expect immediate results. In short-term oriented cultures, the results of the past month, quarter, or year is a major concern. Time is seen in a different way by eastern and western cultures and even within these groupings temporal culture differs from country to country. Also temporal identities of different organizations and teams in organizations may vary

time for meetings and expect that your Japanese colleagues will do the same. Since this is a group society, even if you think you will be meeting one person, be prepared for a group meeting. It may take several meetings for your Japanese counterparts to become comfortable with you and be able to conduct business with you. The Japanese are looking for a long-term relationship. A package of literature about your company including articles and client testimonials are very important. A small gift presented to the most senior person at the end of the meeting is taken as a mark of esteem.

The system can also provide macro-level implications of cross-cultural differences for team working as summarized in Table 5. This interpretation is based on Hofstede's cultural dimensions. As well, the team can study Lewis' cultural model, which is more focused on communication.

5 Conclusions

The last decade has seen a dramatic increase in the importance of cross-cultural collaborative research and development projects both in academic and business life. In our paper, we have analyzed the *characteristics of knowledge*. The purpose of our paper was to point out that the knowledge transferred in SE processes and projects is not free of connections to its environment i.e. *context*. We have discussed the characteristics of knowledge based

on the SECI structure with two types of knowledge—explicit and tacit, and on context level. On the context level of classification, we separated two classes of knowledge—*context-free* and *context-sensitive*. Context-free knowledge is independent of everything that concerns its context (e.g. mathematical rules and formulas). Context-sensitive knowledge is dependent on its context. In this case, in addition to the knowledge itself, its context is also an essential element to be included in the usage situation. We have combined these two knowledge classifications—explicit/tacit and context-sensitive/context-free and proposed an integrated knowledge classification framework. We have applied our classification for constructing the *SE Working Model*. We have also analyzed the problems connected to cross-cultural service processes and software engineering work; this is an interesting subject for further investigations. We have discussed *culture* as one example of context. Integrating cross-cultural knowledge into SE projects and processes is a challenge for developing guidelines, models, methods and practices for the new, *Culture-Aware Software Engineering (CASE)* curriculum. We have proposed that the integration can be realized with our knowledge classification framework. The three-layer model provides a means for a better understanding of the requirements arising from context-dependency and culture awareness for the software engineering process. The Spiral model (Boehm 1988) was selected to describe the development process itself. When connected to other layers, we are able to apply it as a "double spiral" model (SECI spiral inside Boehm spiral steps) or just to analyze the importance of KAs and Bas needed in every step. The third layer binds all this to the cultural aspects to take into account in the application of the selected Bas and specifying the format (language, need for visualization instead of text, etc.) in KAs.

In university level education, culture sensitivity must be taken into account as a part of SE professionals' education. Traditionally, this has been seen mainly in the curricula as foreign language studies, some minor amount of cultural studies, and in SE education in usability oriented studies (Cognitive Psychology, Adaptable User Interfaces,). In a multicultural environment, however, the importance of the use of (semi)formal models instead of natural language avoids problems in communication. Standardized processes—including commonly accepted methods and tools—also supports manageable behavior of a heterogenous teams. The establishment of a new, Culture-Aware Software Engineering (CASE) curriculum in our university alliance (Tampere University of Technology, University of Jyväskylä and University of Tampere) is one of our future tasks.

Acknowledgments The authors thank the Finnish Funding Agency for Technology and Innovation (Tekes) and the Academy of Finland for partly funding our research project.

References

Agile. (2009). Agile Manifesto. http://agilemanifesto.org/. Accessed 14 April 2009.

Ahn, H. J., Lee, H. J., Cho, K., & Park, S. J. (2005). Utilizing knowledge context in virtual collaborative work. *Decision Support Systems*, *39*(4).

Bijl, A. (1995). *Ourselves and computers. Differences in minds and machines*. London: Macmillan Press Ltd.

Boehm, B. (1988). A spiral model of software development and enhancement. *IEEE Computer, 21*(5), 61–72.

Boehm, B. (2006). A view of 20th and 21st century software engineering. In *Proceedings of the 28th international conference on software engineering (ICSE), Shanghai, China* (pp. 12–29). New York, NY, USA: ACM.

Brézillion, P. (2003). Focusing on context in human-centered computing. *IEEE Intelligent Systems, 18*(3), 62–66.
Coutaz, J., Crowley, J., Dobson, S., & Garlan, D. (2005). Context is key. *Communications of the ACM, 48*(3), 49–53.
de Geyetere, T. (2009). A unified model of dynamic organisational knowledge creation. Explanation of SECI model of Nonaka and Takeuchi. http:/www.12manage.com/methods_nonaka_seci.html. Accessed 6 February 2009.
de Mente, B. (2001). *Etiquette. Guide to Japan*. Singapore: Tuttle Publishing.
Dey, A., Kokinov, B., Leake, D., & Turner, R. (Eds.). (2005). Modeling and using context. In *5th International and Interdisciplinary Conference CONTEXT 2005*. Berlin: Springer.
Good, R. (2009). Best online collaboration tools 2009—Robin Good's collaborative map. http://www.mindmeister.com/12213323. Accessed 10 April 2009.
Gorrill, J. R. (2008). Intercultural communication. http://www.kwintessential.co.uk/resources/global-etiquette/. Accessed 14 April 2008.
Heimbürger, A. (2007). Time-contexts in document-driven projects on the web: from time-sensitive links towards an ontology of time. *Information Modelling and Knowledge Bases XVIII, Frontiers in Artificial Intelligence and Applications, 154*, 136–152.
Heimbürger, A. (2008a). When cultures meet–modelling cross-cultural knowledge spaces. In H. Jaakkola, Y. Kiyoki, & T. Tokuda (Eds.), *Information modelling and knowledge bases XIX* (pp. 314–321). Amsterdam: IOS Press.
Heimbürger, A. (2008b). Temporal entities in the context of cross-cultural meetings and negotiations. In Y. Kiyoki, & T. Tokuda, (Eds.), *18th European-Japanese conference on information modelling and knowledge bases* (EJC2008) (pp. 297–315). Tsukuba, Japan.
Heimbürger, A., Jaakkola, H., Sasaki, S., Yoshida, N., & Kiyoki, Y. (2009). Context-based cross-cultural knowledge creation and sharing. In *The Proceedings of the 19th European–Japanese conference on information modelling and knowledge bases.* Maribor, Slovenia.
Hofstede, G. H. (1984). *Culture's consequences: International differences in work-related values*. SAGE.
Hofstede, G. (2003). Geert Hofstede™ cultural dimensions. http://www.geert-hofstede.com/. Accessed 20 March 2009.
Hofstede, G., & Hofstede, G. J. (2004). *Cultures and organizations: Software of the mind: intercultural cooperation and its importance for survival*. New York: McGraw-Hill.
Hoft, N. L. (1996). Developing a cultural model. *International User-Interfaces*, 41–73.
Holden, N. J. (2002). *Cross-cultural management. A knowledge management perspective*. Harlow, UK: FT Prentice Hall.
IEEE and ACM. (1990). Glossary of software engineering terminology (Standard 610-12). IEEE and ACM.
Intercultural Communication. (2009). http://wiki.answers.com/Q/What_is_the_definition_of_intercultural_communication. Accessed 20 March 2009.
ISO/IEC. (2004). ISO/IEC 12207/Amd2:2004—Information technology—Software life cycle processes—Amendment 2 2004.
ISO/IEC. (2006). 15504-5: Information technology—process assessment—Part 5: An exemplar process assessment model, ISO 2006.
Jaakkola H., & Thalheim B. (2005). Software quality and life cycles. In Eder J., Haav H.-M., Kalja A., Penjam J. (Eds.). *Advances in databases and information system. ADBIS 2005, Sept. 12–16, 2005, Tallinn, Estonia* (pp. 208–220). Tallinn University of Technology Press.
Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Wesley: Addison.
King, W. R. (2007). A research agenda for the relationships between culture and knowledge management. *Knowledge and Process Management, 14*(3), 226–236.
Kiyoki, Y., Heimbürger, A., Jaakkola, H., & Takahashi, Y. (to be published in 2009). *Contextual computing of educational knowledge based on Kiyoki's semantic associative search method*. The 20th Anniversary Book of the Information Technology Research Institute at the University of Jyväskylä.
Kiyoki, Y., Kitagawa, T., & Hayama, T. A. (1994). Metadatabase system for semantic image search by a mathematical model of meaning. *ACM SIGMOD Record, 23*(4), 34–44.
Kluckhohn, F. R., & Strodtbeck, F. L. (1961). *Variations in value orientations*. NY: Greenwood Press.
Krishna, S., Sahay, S., & Walsham, G. (2004). Managing cross-cultural issues in global software outsourcing. *Communications of the ACM, 47*, 62–66.
Kroeber, A. L., & Kluckhohn, C. (1952). *Culture: A critical review of concepts and definitions*. The Museum.
Kruchten, P. (Ed.). (1998). *The rational unified process: An introduction*. Wesley: Addison.

Kwintessential. (2009). Cross-cultural solutions. http://www.kwintessential.co.uk/index.php. Accessed 20 March 2009.
Landow, G. (2009). Hypertext versus knowledge management. http://www.cyberartsweb.org/cpace/ht//thonglipfei/nonaka_seci.html. Accessed 6 February 2009.
LeBaron, M. (2003). Cross-cultural communication. beyond intractability. In G. Burgess, H. Burgess (Eds.), Conflict Research Consortium, University of Colorado, Boulder. Posted: July 2003. http://www.beyondintractability.org/essay/cross-cultural_communication. Accessed 27 February 2009.
Lewis, R. D. (1999). *When cultures collide. Managing successfully across cultures*. London: Nicholas Brealey Publishing.
March, R. M. (1990). *The Japanese negotiator*. Tokyo: Kodansha International.
Marcus, A., & Baumgartner, V.-J. (2004). A practical set of culture dimensions for global user-interface development. *6th Asia Pacific Conference, APCHI, 3101*, 252–261.
Marsen, S. (2006). *Narrative dimensions of philosophy*. London: Palgrave.
Nonaka, I., & Takeuchi, H. (1995). *The knowledge creating company: How Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press.
Nonaka, I., & Teece, D. (2001). *Managing industrial knowledge. Creation, transfer, and utilization*. London: SAGE Publications.
Pfleeger, S. L., & Atlee, J. M. (2006). *Software engineering: Theory and practice* (3rd ed.). Pearson Education International.
Polanyi, M. (1967). *The tacit dimension*. New York: Anchor Books.
Scrum. (2009). SCRUM: A team-based framework to develop complex systems and products. http://www.scrumalliance.org. Accessed 14 April 2009.
SEI. (2006). CMMI for Development, Version 1.2. Technical Report, CMU/SEI-2006-TR-008. Software Engineering Institute (SEI).
Shah, S. (2004). The researcher/interviewer in intercultural context: A social intruder!. *British Educational Research journal, 30*, 549–575.
Siakas, K. V., Georgiadou, E., & Sadler, C. (1999). Software quality management from a cross-cultural viewpoint. *Software Quality Control, 8*, 85–95.
Straub, D., Loch, K., Evaristo, R., Karahanna, E., & Strite, M. (2002). Toward a theory-based measurement of culture. *Journal of global Information Management, 10*, 13–23.
Walker, N. N. (2005). Investigating the cognition behind the intercultural interactions of four Japanese teachers of English as a foreign language. *Electronic Journal of Contemporary Japanese Studies*.
XP. (2009). Extreme Programming. http://www.extremeprogramming.org/. Accessed 14 April 2009.

Author Biographies

Hannu Jaakkola is a Professor of Software Engineering at the Tampere University of Technology, Pori. He is the Director of Pori Doctor School and the Head of the Centre of Software Expertise (CoSE). He holds a Doctor of Philosophy (Eng.) degree from the Tampere University of Technology and a Bachelor of Business Administration from the University of Tampere. He also holds a part-time professor (docent) in Information Technology, University of Lapland. He is a Director of the Pori eMBA Programme and the Pori Management Training Programme. He has positions of trust in several international and national scientific organizations and journals. He is the permanent member of the Organization and Steering Committee of the European-Japanese Conference on Information Modelling and Knowledge Bases (EJC) and the co-editor of the EJC Book. His areas of research interest are software engineering methods, software process improvement, quality, object technologies, information society technologies and processes, data mining, data management, technology management, technology diffusion, and technology transfer.

Anneli Heimbürger is a post-doc researcher in the Department of Mathematical Information Technology in the Faculty of Information Technology at the University of Jyväskylä, Finland. She holds a Doctor of Technology degree from the Tampere University of Technology with the specialization in time-sensitive linking structures on the Web, extended XML Linking Language and Allen's temporal interval calculus. The application domains of her Dr. Tech. research are distributed project and knowledge management and usage of technical manuals. She has positions of trust in international scientific organizations and journals. She is currently focusing on temporal information processing, Web continents and levels of linking, and cross-cultural information systems. Her areas of research interest are concept of time and context, culture-sensitive virtual and physical knowledge spaces, and ICT applications to support cross-cultural communication.

Petri Linna is a researcher in the Department of Software engineering in the Tampere University of Technology—Pori. His previous research was a business process model that was linked to emergency management and this moment research areas are cross-cultural theories, dimensions, and collaborative tools that support cultural factors management and software quality management.