Intellectual Capital Based Evaluation Framework for Dynamic Distributed Software Development

Pekka Kamaja1, Mikko Ruohonen2, Katriina Löytty2, Timo Ingalsuo2
1 Haaga-Helia University of Applied Sciences, Helsinki, Finland
2 CIRCMI, School of Information Sciences, University of Tampere, Tampere, Finland
pekka.kamaja@haaga-helia.fi
mikko.j.ruohonen@uta.fi
katriina.loytty@uta.fi
timo.ingalsuo@uta.fi

Abstract: This article presents constructing of an evaluation framework for dynamic distributed software development (DDSD). The topic examines building the capabilities, evaluating the efficiency and scaling up the performance of globally distributed software development teams in environments that demand high operational excellence, innovativeness and other intellectual properties. Three universities and four ICT service and software companies in Finland collaborate on a research project, DD-SCALE (2014-2016). The project objectives are to investigate and develop measurement solutions, tools and work practices for managing and evaluating DDSD work. The challenge of harnessing human and social capital assets for scaling high-performing teams to fit with high-performing organizations is addressed.

The research began with an explorative phase for designing the preliminary concept of the evaluation framework which further defined the research questions. The increased knowledge of the object of study brought a better standpoint to judge among various approaches for the framework. Theories of Intellectual capital (IC), Performance management, productivity and distributed software development were investigated.

The results of the paper are: 1) conceptualizing productivity of DDSD operations in terms of an evaluation framework on individual, team and organizational levels with dynamic IC emphasis; 2) a categorization of evaluation indicators on three aggregation levels; and 3) a baseline construction for the framework with practical trials.
Contributions to the scientific community are: 1) a conceptualization of productivity in knowledge intensive technology developer organizations in terms of dynamic IC and; 2) a model for conceptualizing how the impact of dynamic IC on productivity is manifested and seen in such organizations. Both views extend the applicability of productivity as measurement within knowledge intensive organizations.

Contributions to management practitioners are: 1) management and development of work practices and; 2) guidelines in exploiting the full gain from advancements in high performing software research, development and innovation (RDI) within globally distributed setting.

Keywords: dynamic distributed software development, global software development, distributed teams, software evaluation, intellectual capital, performance management, knowledge work productivity

1. Introduction

Outsourcing information technology (ITO) and business and knowledge processes (BPO) has increased tremendously during the last two decades and now provides great business opportunities for many knowledge-intensive companies (Lacity and Willcocks, 2001; Lacity, Willcocks and Cullen, 2008; Saxena, Ruohonen and Bharadwaj, 2010). Since the rise of outsourcing business in the US and Western European countries in the late 1990s, the business today is ruled by the offshoring offerings in cost competitive countries of Asia, Latin America and Africa (Ruohonen, Mäkipää and Kamaja, 2014). Motivation for distributing RDI work relates to sustaining contacts with customers at remote locations, exploiting the availability of remote workers, reducing costs by offshoring, and enhancing the capabilities by creating networks with other development organizations and teams (Fuggetta and Di Nitto, 2014).

Distribution of the RDI operations of software development companies is being challenged by the complexity of the combination of onsite, onshore, nearshore and offshore settings (Oshri, Kotlarsky and Willcocks, 2007). Consequently, one-way outsourcing operation that hands over assets, people, activities and knowledge to third-party management does not remain competitive anymore. Two-way, collaborative and network-based contracting that constantly evolves can release a company's knowledge potential and simultaneously release the provider's potential, resulting in mutual gain (Ibid.). Thus, in a networked business environment, it has turned to be inevitable to make relation-based business operations that may also be-come contributed by spontaneous collaboration and social networking besides the more planned and man-aged practises (Begel, Herbsleb and Storey, 2012; Dabbish, et al., 2013).

Currently cost competitive countries are able to compete on the market of software engineering resource due to their cheaper labour costs. In particular, service levels, dynamic competencies and community-based activities can act as game changers when evaluating total costs. However, human capital, human resource management complexities and various administrative issues can result in unfavourable events and unexpected situations when managers are attracted by cost savings brought with offshoring opportunities (Rottman and Lacity, 2006).

Due to the various forms of outsourcing, management of such networks is challenging, in which not only information technology (such as servers in clouds), but also business and knowledge processes are being distributed through the whole network. The increased complexity in networked operations followed by care of cost effectiveness can be met with enhanced transparency and synchronization across the distributed software development (Herbsleb, Kastner and Bogart, 2016; Dabbish, et al., 2013; Cataldo and Herbsleb, 2013).

Considering cost efficiency, surprisingly many of software vendor companies undertaking offshoring projects constitute their performance monitoring both on overly simplified and blurred yardsticks in measuring the performance of their external outsourcing partners (Rottman and Lacity, 2006). Moreover, "[t]he productivity of knowledge intensive organization in terms of knowledge management performance monitoring is situational and context dependent" (Johnson, Mawson and Plum, 2014). Finally, "[t]here is no standard or single, widely acknowledged metric, method, or set of key performance indicators for measuring the more complex forms of knowledge worker productivity" (Ibid.).

In search of cost savings and higher productivity, the dream of "to work when and where people prefer to work using fast and mobile IT-facilities" (Gorgievski, et al., 2010) has now become trivial. In near future, the new ways of working in the ICT (Information and Communication Technologies) industry and other knowledge work organizations are influenced by the increased role of computerization. Computerization of knowledge work in terms of artificial intelligence and machine learning solutions has been seen "augmenting the work of highly skilled labour, while allowing some types of jobs to become fully automated" (Frey and Osborne, 2014). In software engineering, the objects of future computerization are the non-routine cognitive tasks predominantly characterized by pattern recognition (Brynjolfsson and McAfee, 2011) like the optimization of complex design choices (Hoos, 2012 cited in Frey and Osborne, 2014, p.15) and more advanced software bug detection (Frey and Osborne, 2014). Thus, information technology is shifting forcefully from a servant role towards more challenging intelligent tasks in knowledge work.

The key rationale behind the DD-SCALE RDI program (2014-2016) is managing distribution of software development work. This research was conducted together with two case companies that operate in software RDI intensive industries. Both of the companies have several sites distributed globally. Besides the two, other two partner companies in the program participated in project meetings and commented on the findings, which contributed to this study.

The software companies' decision makers were bothered especially by how to judge total productivity, which is predominantly dependent on the productivity of software engineering teams and developers. Ex-tending the knowledge and understanding of the underpinning causes that explain productivity, together with the aim for a comprehensive evaluation framework that could be used for various purposes by managers of RDI-operations, were the primary interests for the research.

At the outset of the DD-SCALE program, a work hypotheses and problem statement was that the primary objective of the framework would be to explain productivity of RDI-operations in software engineering companies. The goal was to shed more light on the impact of distribution on productivity. Other views included finding measurements for comparing total efficiency across various sites of a company, measuring the impact of a company's organisational changes, and estimating the impact in work transfer across company sites.

The problem statement is interpreted in the form of three research questions (RQs): RQ1: What are the applicable dimensions, in the context of IC, of a comprehensive and scalable evaluation framework for DDSD? RQ2: What are the relevant indicators for evaluating the performance of DDSD? During the research process, RQ2 was directed towards explaining performance and/or productivity of distributed software development work. And finally, RQ3: How can a framework, its dimensions and indicators be effectively implemented in practice?

2. Theoretical consideration

Theoretical consideration is divided into two main sections. The first section defines the design criteria for constructing the evaluation framework. The second section defines the dimensions involved in distributed software development that eventually conceptualize the framework from the theory point of view.

2.1 Design criteria for the framework

Software evaluation approaches like COCOMO, SLIM and Price-s in the 80's (Reifer, 2007) and their enhancements in the 2000 millennium to meet the requirements set by Agile approach are about quantifying productivity of work in terms of cost, effort and duration that are the inputs expended to produce the output (Ziauddin, Shahid and Shahrukh, 2012). The more contemporary evaluation approaches, according to the agile way, either using metrics such as story points, object points or use case points are meant for monitoring work left to do versus time in a given timespan (Huskins, Kaplan and Krishnakanthan, 2013). Common for all of them is the use of direct inputs and outputs that fall short in explaining the less direct factors sur-rounding the immediate inputs by developers.

Moreover, software engineering like any knowledge intensive work is "the creation, distribution or application of knowledge by highly skilled (and autonomous) workers using tools and theoretical concepts to produce complex, intangible and tangible results" (Bosch-Sijtsema, et al., 2009), which is aligned with the characteristics of software development work. Consequently, *the first design criterion is that the impact of less direct tangible and intangible inputs are the parameters of the DDSD evaluation framework.*

"Knowledge worker productivity should be assessed on the team level, because knowledge work is not an individual task, but usually performed in collaboration with others on complex tasks that they cannot perform alone" (Johnson, Mawson and Plum, 2015). Therefore, besides the focus on productivity of individuals, the team level is necessary (Ibid.). Bosch-Sijtema, et al. (2009) suggest that in addition to the individual level, which is knowledge workers in different work modes and tasks and team level, the work environment (physical, virtual and social workspaces) and organizational level (organizational context) are feasible levels in measuring knowledge work productivity.

The complexity of measuring productivity in knowledge intensive arrangements can be found in literature from the early 90's with the question of IT-investment productivity, known as productivity paradox (Brynjolfsson, 1993). The hope for short-term gain within the boundaries of an investor company was overruled by the study which stated that IT-investments made by supplier industries increase the productivity of downstream industries (Han, Chang and Hahn, 2011). Thus, the horizontal dimension in defining the productivity factors spans beyond the organization boundaries.

Therefore, the second design criterion is that the individual perspective needs to be complemented by the levels of distributed teams, business lines/units and organizations. Moreover, the coverage must be stretched beyond the company boundaries.

To meet these two requirements, the performance management approach could be an adequate choice. Performance management frameworks are crafted with leading and lagging perspectives that enable extending the cycles of monitoring much broader than that addressed in productivity measurement approaches. For example, one cycle is organizational learning, including team based learning, that eventually enables individual workers' higher performance. Although performance measurement systems are powerful in monitoring the overall performance of organizations and linking the measurements to strategy, they lack assessing the individual perspective (Jääskeläinen and Laihonen, 2013). Moreover, they are susceptible to biases that can be seen in the challenges of linking key performance indicators with business operations, such as software engineering (Reddy and Ryman, 2009). They also fail in exposing the under-pinning root causes behind the measurement parameters, unlike Intellectual capital which is linked to the organizational competences and processes in subtler way (Lerro, 2014).

One branch in performance management is knowledge work performance measurement. Frameworks are rich in explaining the individual, team and organizational knowledge work related assets. One quite a new framework by Palvalin, et al. (2014) holds two main sets: 1) drivers related to work environment and employees' ways of working, and 2) the resulting factors related to well-being at work and productivity. In most of these appraisal tools, the focus is on Human and Structural Capital (Johnson, Maw-son and Plum, 2015) but also signs of Relational capital can be found, like the consideration of customer perspective (Palvalin, et al., 2014; Xiao, Nembhard and Dai, 2012).
Accordingly, the third design criterion is that the evaluation framework shall comprise a cyclical approach that enables capturing the intangible capabilities holding diverse cycles of impact.

The notion of capabilities is found not only in performance management literature, but also in Intellectual capital. Although at first the financial accounting driven static viewpoint treated company's intangibles as stocks of assets, further research extended this view. Indeed, dynamic Intellectual capital (Leitner and War-den, 2004; Ståhle and Grönroos, 2000) was seemingly inspired by the resource based view, where the capabilities could be found as a unit of analysis in explaining the competitive advantage of companies (see e.g. Dierickx and Cool, 1989; Kogut and Zandler, 1992).

The valuable aspect in performance management frameworks is that they are dynamic and capture the various cycles of intangibles impacting on overall performance. For example, the Balance Scorecard -framework captures three major forms of intellectual capital, Human, Structural and Relational capital in the perspectives of Learning and growth, Internal process and Customer – all of them bearing an impact on company progress, especially increasing the shareholder value (Kaplan and Norton, 2004).

The estimates for the impact cycles of Intellectual capital are as follows: 1) Human capital and organizational knowledge that is the learning and growth perspective in building future growth 3 – 5 years; 2) Structural capital in terms of internal process management, productivity and cost-efficiency 6 – 12 months and; 3) Relational capital in terms of customer service, satisfaction and quality of service related perspective 12 – 24 months (Ali-Yrkkö, 2008, Neely, et al., 2002, Shenhar, et, al., 2001 cited in Kamaja, 2012). The longest one is the impact cycle of learning. As it occurs on organization level, it does not reflect the impact cycles on individual level. Learning new skills in software engineering industry such as new programming language would grant relatively quick wins within one-year time frame, but a comprehensive shift into new technologies at company level would require 9 – 18 months to realize the full benefit.

Finally, the fourth criterion is that dynamic intellectual capital approach shall enable linking fragments of intellectual capabilities with chosen output metrics.

The chosen output metric in this study is productivity of software engineering operations. Productivity in general is defined as the ratio of output and input. In software engineering, "the output represents the out-come of the process, which can be the product artefacts, the documentation, or the value of the outcome" (Cheikhi, Rafa and Ali, 2012). The meaning of productivity varies depending on the context. Consequently, productivity takes a different perspective on each management level (Tangen, 2005). A more sophisticated definition with Lean management emphasis by Slack, et al. (2001 cited in Tangen, 2005) nails the five components of high performing operations: High-quality (do not waste time or effort in re-doing things); Fast operations reduce the time to market; Dependability of operations; Flexibility to adapt to changing circumstances quickly. The fifth is low cost of operation that escapes the scope of Intellectual capital.

Definitions for making sense of productivity are also taken from the Intellectual capital perspective (Bontis, 1999). Human capital is linked with volume on individual tasks. The higher the volume is the faster the operations are. The essence of structural capital is cultivating internal routines, ways of doing tasks, which focus on efficiency and accessibility. Relational capital resonates with longevity, that is sustaining the relation-ships (Bontis, 1999, pp.445-450). Table 1 summarizes the concept of productivity cross-referenced with Intellectual Capital.

Table 1: A summary of the concept of productivity cross-referenced with Intellectual Capital

Essence *)	Human Intellect	Organizational	Business relationships
Scope *)	Internal to employee	Internal within Organization	External to the organization
Parameters *)	Volume = throughput and quality	Efficiency, Accessibility = Integrity, Cohesiveness, Smoothness	Longevity = Expansion/Growth
Categories **)	Unstructured human knowledge and skills Motivation Learning and Renewal Social/Bonding	Structured and shareable knowledge Organisational structures, policies, processes Organizational empowerment (leadership issues)	Customer relationships Relationships with Partners/Collaborators

Essence *)	Human Intellect	Organizational	Business relationships
Productivity parameters	Fast operations, Avoiding waste of time	Dependability and Flexibility as they are management related topics	Flexibility needed to satisfy changed customer preferences
Common terminology in software engineering context	Velocity	Quality	Appropriateness (to customer)

*) Bontis, 1999 pp. 445-450

**) Huang, Luther and Tayles, 2007; Cricelli, Greco and Grimaldi, 2014

As a conclusion here, in the software engineering context, productivity is defined as velocity, quality and appropriateness to customers which are in line with the concept of Human, Structural and Relational Capitals, respectively.

2.2   Facets of the dimensions in the framework

The landscape for discovering new tools for managing DDSD (see Ruohonen, Mäkipää and Kamaja, 2014; Kamaja, Ruohonen and Ingalsuo, 2015) is framed by the search for desired benefits, such as savings in cost and delivery time, securing IT manpower and achieving market proximity (see Dutta and Roy, 2005). This also applies to secondary objectives, such as inducing innovativeness (Kojima and Kojima, 2007). The counterforces acting against the benefits of the distributed management models are poor communication, such as gaps or unclear chains of command; cultural differences; the transferring of the business domain; decreases in project visibility; configuration management; a disconnect between project estimates and feasible results; client business security; document maintenance and synchronization (Hameed and Nisar, 2004).

The ideal solution for the purposes of DDSD would be to provide a broad and deep analysis approach. The breadth of the analytical framework is due to the aforementioned challenges, but especially to the main categories and the factors that are present on the level of distributed teams (see Kamaja, Ruohonen and Ingalsuo, 2015; Löytty and Ingalsuo, 2015):

1.   Cross-cultural factors (Fontaine, 2007; Hudson, 2007);
2.   Organizational values and leadership (Schein, 2010);
3.   Communication in and between teams (Sahar, Raza and Nasir, 2013);
4.   Remote collaboration patterns between teams (Herbsleb and Mockus, 2003) and
5.   Knowledge management (Oshri, Kotlarsky and Willcocks, 2007).

The first main category, the cross-cultural perspective can be seen in two ways: 1) in the working of the multisite organization, and 2) in the working of multicultural project teams. Multicultural teams have a higher potential for greater success than single-culture teams do, but they also have a higher risk of failure. Cultural differences in project management can be difficult to navigate, especially in the software industry. (Hudson, 2007). It is important to acknowledge the importance of cultural competence. A good starting point for in-creasing cultural competence is offered by different cultural typologies. The advantage of these models lies in their power to make sense of a different culture, even if the person using these models does not have first-hand experience of the specific culture. The thorough consideration of the next main dimension, organizational values and leadership would require a more precise investigation of the underlying factors. House, et al. (2004) have presented nine dimensions of leadership which are in line with the five factors presented by Hofstede (2001). Other relevant taxonomies explain the cultural aspects involved in leadership (Trompenaars and Hampden-Turner, 1998) and the cross-cultural aspects of managerial work (Jacob, 2005).

Remote collaboration patterns are taken into use when managing the knowledge and the division of work between different sourcing sites. When the division of work is based on expertise, it utilizes the knowledge and expertise of a company's employees regardless of their geographical location. Thus it allows these companies to access the pool of expertise available in offshore locations, where the familiarity between peers and knowing their expertise profiles is pivotal (Marlow, Dabbish and Herbsleb, 2013). Lastly, an expertise-based division of work approach requires that remote engineers and managers interact, and consult with their counterparts in order to solve design issues. Kotlarsky, et al. (2007) observed that companies which attempted to reuse components across different projects and products, and improve product flexibility through the application of component-based development were especially dependent on the success of a) inter-site coordination; b) knowledge management and; c) communication channels. Also a sound product architecture that reduces technical dependencies enables efficient inter-site coordination and,

furthermore, a more efficient distribution of engineering resources by reducing work dependencies between teams (Cataldo and Herbsleb, 2013). Also the dimensions of customer orientation and business models have an influence on operations (Oshri, Kotlarsky and Willcocks, 2007). In addition, Herbsleb and Moitra (2001) argue that cultural communication and knowledge management issues are significant factors.

A quite recent set of categories developed by Prikladnicki and Audy (2012) that focus on the DDSD field are probably best able to link the IC tradition. The categories are distance, levels of dispersion, organizational structure, the practices of operations, culture, trust, collaboration patterns, the division of project work across sites, development methods, policies and standards, the measurement of the productivity of distributed software development and project management and leadership. Klein (2008, p.2) suggests this organizational culture and leadership is capital, whereas Bontis (1999, p.450) sees them as external to the drivers of intellectual capital. Linking intellectual capital to globally distributed software engineering is troublesome, not only due to the inconsistencies in the categories and their concepts, but also because of the differences between the objects of investigation. Distributed software development is anchored in the phenomena of the global software engineering context, whereas intellectual capital is interested in intangible assets. These categories and the related factors that emphasise the essence of distribution management are here considered as the lenses for ascertaining the ontologies residing in the area of interest (Kamaja, Ruohonen and Ingalsuo, 2015; Ruohonen, Mäkipää and Kamaja, 2014).

To summarize the theoretical discussion so far, it is evident that the literature discusses the categories and factors present in DDSD work in different ways and with a variety of perspectives and emphases. In an attempt to gain a unified view of the different factors that impact on the collaboration and productivity of distributed teams, the first steps taken in the DD-SCALE program were to create a multi-layered concept map based on a literature review (Löytty and Ingalsuo, 2015). The key findings were that distributed team collaboration and productivity are surrounded by various elements originating from different levels in relation to the team (Espinosa, et al., 2007). The temporal, physical and socio-cultural distances often inherent in distributed teamwork have an important influence on the factors at the team, organizational and operating environment levels. These factors come closer to the core of doing the actual work. If successfully man-aged, the factors can support the collaboration and productivity of the teams, but if lacking or misdirected, they can effectively act as hindrances (Espinosa, et al., 2007; Johnson, Mawson and Blum, 2015, Löytty and Ingalsuo, 2015).

3. Research approach, data collection and analysis

The overall research approach is design science and action research that has a qualitative emphasis. Design science "creates and evaluates IT artifacts intended to solve identified organizational problems" (Hevner, et al., 2004, p.77). "Central to action research is collaborating, co-creating solutions and crafting new ways of operating together with project stakeholders", such as the case companies (Atweh, Kemmis and Weeks, 1998).

The research process can be characterized by three main perspectives: First, literature in Performance Management and IC disciplines provided the theoretical foundation for defining the key concepts needed in constructing the evaluation framework. They also framed the scope of research. Second, problem domain specific literature of distributed software engineering and global software development were applied in formulating the context dependent data collection plan. The concepts available in this literature were also particularly useful in conceptualizing the framework. Third, data collection in the case companies was carried out by methods of interviews and workshops.

The data was collected in semi-structured interviews and theme based workshops in the case companies between January and August 2015. The themes for the interviews and workshops were identified based on the literature review in the chosen disciplines and specification meetings within the DD-SCALE project. The interviews and workshops examined the topic from different angles with the aim to cover a broad array of perspectives to the topic and establish the dimensions and potential indicators for the evaluation frame-work. The data collection included both individual and group sessions. The informants were from Finland, India and Malaysia.

The 16 transcriptions of the data collection sessions (interviews and workshops) were analysed by qualitative content analysis. The aim was to identify and conceptualize the relevant phenomena for assessing productivity in DDSD work: textual data was analysed by seeking for and categorizing meaningful entities related to productivity and high performing software engineering work. (Schreier, 2014) The analysis process was iterative and collaborative among the researcher team. During the process, interim results were also presented to and discussed with case company representatives.

The analysis was conducted with the help of Atlas.ti -software that facilitates coding, categorizing and man-aging the collected research data. The analysis moved from raw coding the data towards a more fine-tuned, focused and categorized set of elements. Throughout the analysis discourse was carried between the raw data, nascent results, theoretical underpinnings and researchers' feedback. In other words, "the process of interpretation move[d] from a precursory understanding of the parts to the whole and from a global understanding of the whole context back to an improved understanding of each part" (Klein and Myers, 1999).

Through this conduct, a set of raw factors related to productivity in DDSD work, that is, indicator candidates, were established. These candidates were then critically reviewed by the researcher team, in order to identify potential duplicates and other redundant elements that should be excluded from the results. In parallel with this, the indicators were categorized following the principle that "categories are higher level and more abstract than the concepts they represent" and that a category must reflect the properties, dimensions and characteristics of the phenomenon it aims to represent (Corbin and Strauss, 1990). Also the recipe for a useful taxonomy was found helpful in this instance: a taxonomy should be *concise, robust, comprehensive, extendable* and *explanatory* (Nickerson, Varshney and Muntermann, 2013). Figure 1 shows a snapshot ex-ample of the coding work to illustrate the analysis process with the steps of the examples explained in the top-right corner.
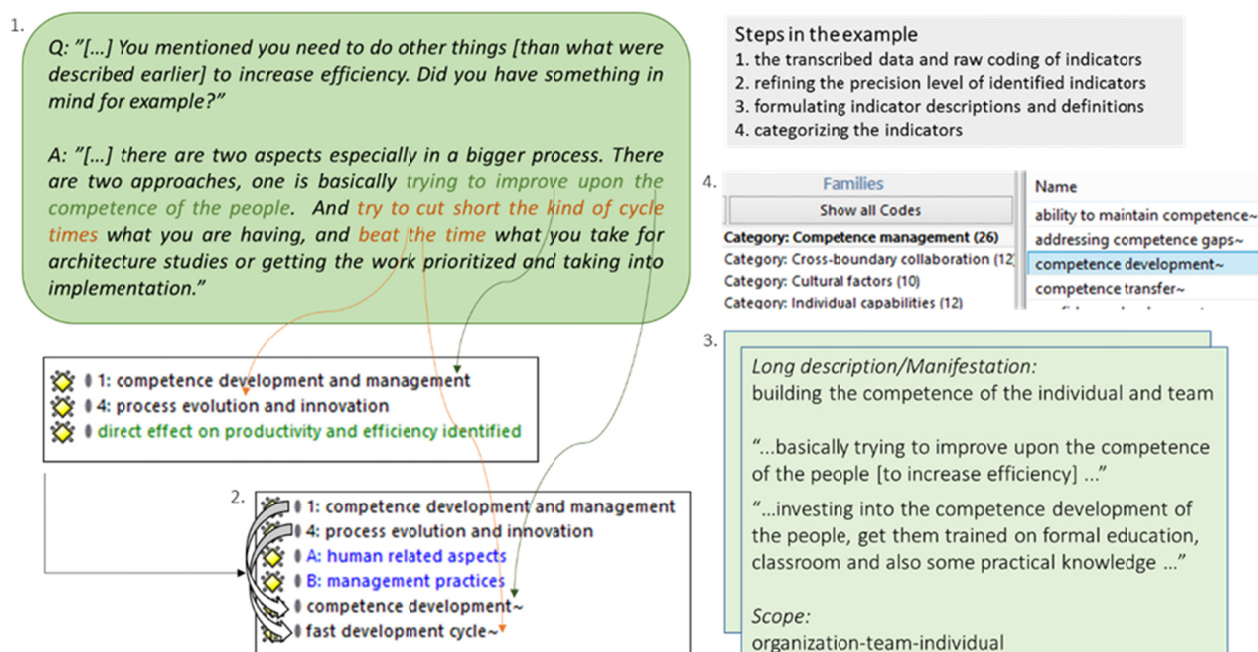

Figure 1: A snapshot example of the coding work during the analysis process (applied from Löytty, 2016)

Intellectual capital was taken as one of the main approaches in this study, especially as it enables breaking down the capabilities of organizations to grass roots. In its well-known form, Intellectual capital is defined as Human capital (HC), Structural/organisational capital (SC), and Relational/Customer capital (RC) (Brooking, 1996; Edvinsson and Malone, 1997; Sveiby, 1997). Other more contemporary frameworks divide Intellectual capital in more a precise way. Jacobsen, Hofman-Bang and Nordby (2005), suggest that HC is divided into the abilities of management and human resource capabilities, SC is divided into innovation and internal pro-cess capabilities, RC splits into networking capabilities and customer loyalty. Another industry specific sys-tem, Deutsche Schmalenbach Gesellschaft fur Betriebswirtschaft eV (DSG) addresses seven subcapitals: human, customer, supplier, process, innovation, location, and investor capitals (Gerport, 2008 cited in Kamaja 2012). These approaches are troubled by the role of social capital as it is pervasive throughout the human, structural and relational capitals (Kamaja 2012). And yet, the intellectual capital categorizations could be more rich entailing even some 70 subcategories (Cricelli, Crego and Grimaldi, 2014; Huang, Luther and Tayles, 2007).

For this study's purposes, the traditional triplet of IC was applied. The three dimensions were defined: Hu-man capital explains the human capabilities, such as knowledge, skills, motivation, learning, social abilities. Structural capital is about organizational structures, structured knowledge and practises and processes. Relational capital consists of

intentional business relations, company external networks and brand. The suggested more detailed IC categorizations were primarily used to assist in the mapping of appropriate categories for the indicators. Figure 2 presents a composition of elements that were finally identified during the analysis.
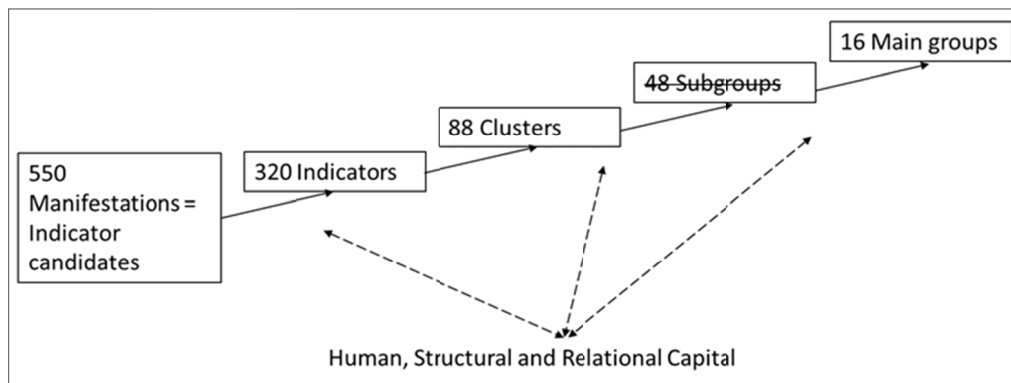

Figure 2: Levels of categorization of indicators

Initially, 550 indicator candidates were identified, which were reduced to 320 indicators after removing duplicates and other non-relevant items. These indicators were categorized into 16 main groups related to the chosen three dimensions of Intellectual capital. The indicators were aggregated into 88 clusters of closely related items within the main groups. Naming the clusters was carried out by using inductive reasoning and crafting the clusters with the most appropriate name that highlighted the nature of the related indicators. After some iterations, the concept was frozen and finalized. The level of subgroups was initially utilized in the categorization work, but it was later removed as redundant.

Although naming the categories was firstly assisted by terminology taken from the intellectual theories, it was modified during the iteration rounds to better match the problem domain, which aligns with principles of grounded theory (Corbin and Strauss, 1990). Because of this, eventually the clusters substituted the level of subgroups, which was initially created to assist in the classifying job.

4.   Results

Along the project, the research team carefully considered the used terminology: The theoretical consideration of means and ends or the causes and effects, needed to be articulated in an established manner. Capability as the unit of analysis turned out to be the most robust concept. Practical arrangements, such as enterprise architecture planning, define capability comprising of three dimensions: people, processes and materials. The first two make sense with human and structural capital. Materials are defined as infrastructure, IT and equipment which resembles the definition of structural capital, too (The Open Group, 2009, p.351).

The measurement of organizational performance (Parmenter, 2007, p.9) is concerned with three types of indicators: Lagging indicators that are the key result indicators. Performance indicators tell what to do right now. Key performance indicators (KPIs) tell what to do to increase performance dramatically, which are the leading indicators. KPIs share somewhat the idea of the DD-SCALE indicators as leading indicators. However, one of the differences is that the DD-SCALE indicators at grassroots are highly detailed entities and they require to be aggregated when used for measurement purposes. Also by nature, they express how a particular activity is enabled, a kind of description of the requirements for sustenance of actions.

An illuminating example of the relationship of KPIs and the DD-SCALE indicators is how software companies monitor bugs, faults in software. The count of bugs is measured typically in three stages within the overall process of developing software code. The first stage is the detection of faults in the actual software programming phase. The second is the code testing phase. Finally, the third and least desired phase to discover faults from the software, is at the customer premises after implementation. From these three KPIs the second, fault detection in testing phase is linked with multiple indicators. In the front line in explaining this particular KPI are three identified human centric capabilities: core software skills, quality care skills and in certain extent also self-reflection of one's skills level, the last denoting one's ability to judge his/her skills relative to the required skill-level of the task. Also the team-level entities make sense here, especially the collaboration and tools related capabilities. Documentation would also help in finding

bugs. These capabilities are manifested on the cluster level of the categorization (Figure 2), here referred to as Capability Indicators.

The overall breakdown structuring of the results is seen in Figure 3 below with a detailed example from one of the main groups, *Team interfaces and collaboration*.
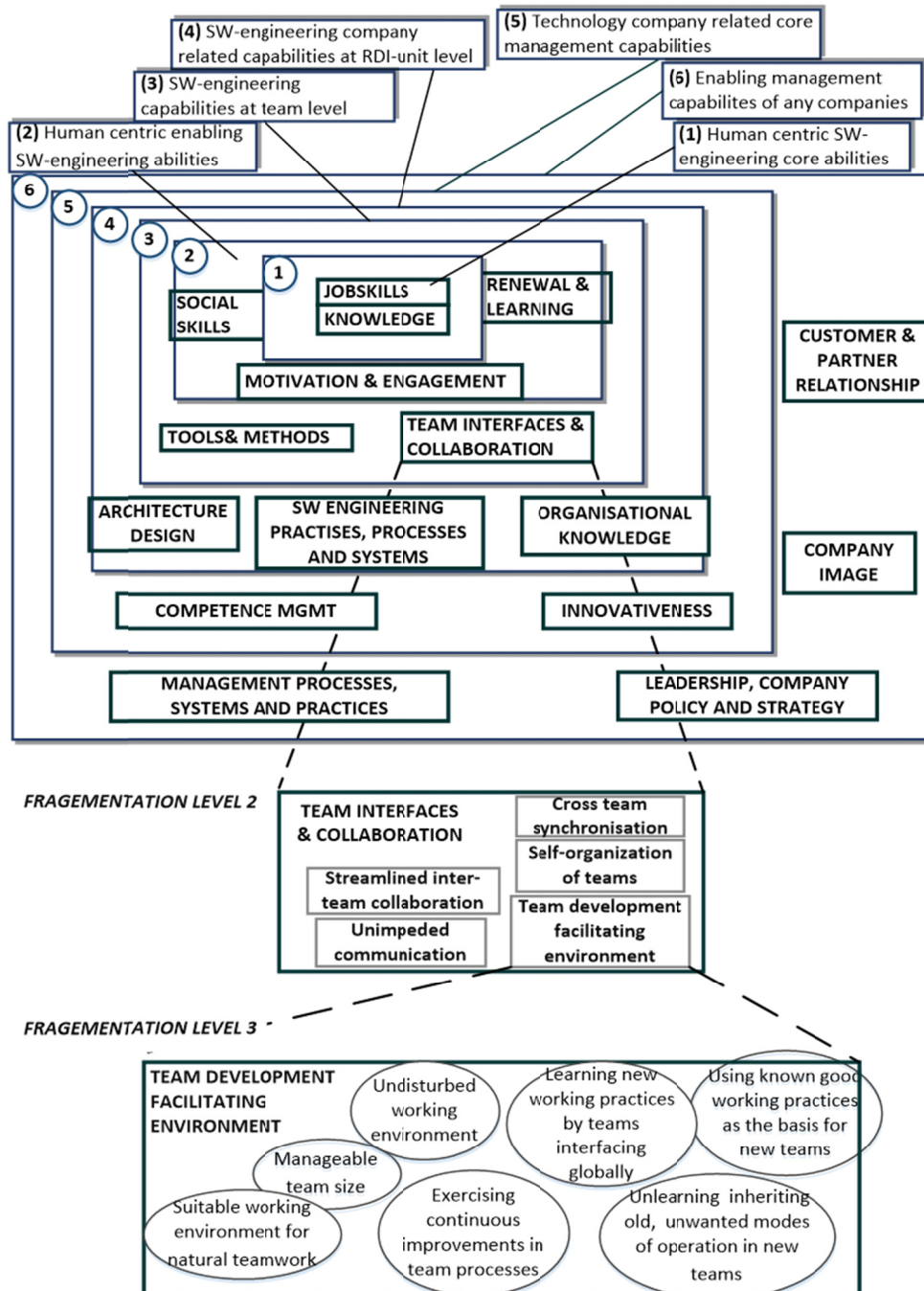

Figure 3: Capability indicators breakdown structuring with the six immediacy levels of productivity of intellectual capital

The main body of the figure, with the six nested boxes and their descriptions in the numbered text boxes above, depicts the levels of immediacy of the categories to productivity impact. The impact of a particular capability to productivity is seen to be the more direct, the higher it is located in the figure. Consequently, the capabilities relating to Job skills and Knowledge are the front line entities in explaining the overall productivity of distributed software engineering work and vice versa, while the capabilities in the main group of Leadership, company policy and strategy are among the least direct. However, the capabilities on each level are strongly interconnected two-way.

Viewing the levels in more detail, level 1 contains the human related core capabilities, i.e. job skills and knowledge, which are in the very core of software engineering and highly crucial to successful work performance as well as to productivity. Level 2 entails social skills, renewal and learning, as well as motivation and engagement which are enablers for the core capabilities on level 1. On level 3 are the team related capabilities, team interfaces and collaboration as well as tools and methods that are crucial to the success of teams. Level 4 refers to software engineering specific management practices and architecture of product technology which is highly contributory to managing the distribution of resources (Cataldo and Herbsleb, 2013). Level 4 also holds structured organizational knowledge in its various forms (artefacts, documents, etc.). The level 5 is about the core of management capabilities seen in technology companies, which are the innovativeness and competence management central to successful software development. Lastly, the level 6 holds the general management and leadership capabilities as well as care of customers and company image that are fundamental building blocks of any company.

The illustration in the lower part in Figure 3 drills down in the categorization from the main group level to the clusters and finally to the individual indicators. The exploded view presents one of the main groups, Team interfaces and collaboration: The fragmentation level 2 shows the five clusters within that main group. The fragmentation level 3 then illustrates the seven indicators within one of the clusters, *Team development facilitating environment*.

Interpretation of productivity can be exercised on the levels of individual indicators and clusters. For in-stance, one of indicators within the framework is *Established cross team connections for quickly accessing expertise*. Undoubtedly getting help outside the team is crucial for the team performance to continue their work in a troublesome situation and, moreover, avoid a stoppage that could negatively influence one of the main productivity measures, namely velocity. Similarly, each of the 320 indicators on the third level in the decomposition of the evaluation framework have either direct, somewhat direct or indirect impact on the object of doing, software engineering.

This example is a manifestation of how to connect dynamic intellectual asset fragments with productivity yardsticks. Although this task is burdensome and complex, it is viable. Instead of analysing all indicators one by one, the level of clusters, one level up, turned out to be homogenous enough to define the related indicators of particular cluster equally in terms of how they impact on productivity. This notion was due to the clusters being of same quality of intellectual capital.

Coming up to the level of the main categories, the intellectual capital value adding to productivity, divided into the intellectual capital impact cycles can be found quite clearly. However, the interpretation here requires the subcategories of human, structural and relational capital. Yet, a detailed disclosure of cycles is not possible here. Instead, discussion on a general level is taken next.

The two main groups, Knowledge and Job Skills, on the first level holding 12 clusters, are the front line capabilities, Human capital by nature, that are the key to high overall productivity. The three main groups on level 2 are also characterized by Human capital. Together they include 15 Human capital capabilities that enable the effectiveness of contributions by individual developers. Furthermore, these are supported by the team level capabilities on level 3, 11 capabilities in total. The level 3 main groups are characterized by Structural capital that can be seen in less or more structured forms, like collaboration patterns or communication practices.

Level 4 is occupied by three main groups, organizational knowledge, product architecture design and soft-ware engineering related process and practices, all them belonging to Structural capital. Sound product architecture design enables efficient division and distribution of design work and is structural capital, too. The 21 capabilities in the three main groups on this level enable efficient team working, which are the level 3 team focused capabilities.

Level 5 capabilities, innovativeness and competence care at organization level, are crucial especially for technology companies and supportive to RDI operations in general. Level 6 contains the general capabilities forming the foundation to management operations and company leadership. They belong to Structural capital although there are IC frameworks arguing of the role of innovativeness as one of the main intellectual capital categories, additional to the traditional triplet (Gerport 2008 cited in Kamaja, 2012). Finally, level 6 includes the Relational capital elements of company image and customer and partner relationship.

5.   Discussion and conclusions

The rationale of the study was to gain a comprehensive and in-depth view for understanding productivity underneath the tangible level of daily software engineering work. More generally, extending the concept of productivity within distributed knowledge intensive organizations, such as software engineering companies was exercised. In practice, the research discovered the ultimate entities at the grassroots level, that is, the indicators. Eventually, a rich and encompassing categorization of the indicators related to the causes of overall performance of daily work in software companies was created.

The DDSD framework is now defined as an evaluation solution with yardsticks to investigate productivity in software engineering work. In reference to the 88 clusters, the concept of Capability Indicators, is one of the key findings of this study together with the yardsticks to better explain productivity in distributed software work. The discussion part is divided into the perspectives of framework itself and its practical uses.

The first perspective is the value adding. Value adding in cycles, can be seen taking effect from the sixth to first level although the capabilities on the different levels are impacting two-way. For example, the individual level capabilities have impact on team level capabilities, too (Kamaja, 2012). However, this study suggests that the dominant direction of the value stream is from general organizational level capabilities up to the individual software engineering working capabilities through the four other levels.

The study also suggests, that the six levels in the DDSD evaluation framework represent the diverse cycles of intellectual capital impacting on productivity. However, the span of the impact cycles is not defined here. The first two levels (1 and 2) are characterized by human capital which are both core and enabling capabilities in gaining higher productivity. The next two levels, team related (3) and software engineering specific (4) capabilities are forms of structural capital, the practices of communication within the organization boundaries, organizational structures and processes and structured knowledge. The level 5 contains technology specific capabilities that are also structural capital. The level 6 is two-fold holding both corporation level structural and relational capital.

The effectiveness point of view, how greatly a particular capability impacts on overall productivity, is not only dictated by the position on the six levels of value adding stream, but also the size of the related entity should be taken into account. For instance, change in architecture may result in better distribution of re-sources involving $50 - 100$ experts, whereas learning to use a new and more efficient programming tool would not necessarily impact more than a couple of employees. Consequently, although a particular capability has a lower immediacy (on the scale of $1 - 6$) and it respectively embodies a less direct impact on overall productivity, the impact must be adjusted by the magnitude of the effected size of organization.

Making sense between the indicators (320) and their aggregates, the clusters (88) becomes more understandable through their practical uses: Indicators can be taken as a starting point in deriving feasible questions for surveys within software engineering, while the clusters are the categories of questions. Moreover, unlike the individual indicators, the clusters denote the operational entities at the most fragmented level in investigating the capabilities of software companies.

Several practical uses for the DDSD evaluation framework were identified by the case companies. For ex-ample, the indicator set can be used as a check list for improving daily operations. It can also be utilized in drafting surveys, for instance, on the impacts of shifting towards a new operational model in managing software teams. Moreover, the transfer of development work from one site to another encompasses risks and uncertainties which can be analysed with the support of the suggested framework. A chosen indicator set can also be used in estimating the current indicators of productivity of the original site and the estimated levels in the destination site after a work transfer. Eventually, figures could be given to all selected sets and their productivity parameters in order to allow a comparative estimation.

Finally, the common aspect for all of the planned and envisioned uses of this baseline model and its derivatives is monitoring change, instead of trying to capture absolute figures. Hence, a comparative measurement approach is amongst the first further plans for the research.

Acknowledgments

References

Atweh, B., Kemmis, S. and Weeks, P., 1998. *Action research in practice*. London: RoutledgeFalmer.

Begel, A., Herbsleb, J. M. and Storey, M-A., 2012. The Future of Collaborative Software Development. In: CSCW '12, *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. Seattle, Washington, USA, February 11–15, 2012.

Bontis, N., 1999. Managing organisational knowledge by diagnosing intellectual capital: framing and advancing the state of the field. *International Journal of Technology Management*, 18(5/6/7/8), pp.433-463.

Bosch-Sijtsema, P.M., Ruohomäki, V. and Vartiainen, M., 2009. Knowledge work productivity in distributed teams. *Journal of Knowledge Management*. 13(6), pp.533-546.

Brooking, A., 1996. *Intellectual Capital: Core Assets for the Third Millennium Enterprise*. London: International Thomson Business Press.

Brynjolfsson, E., 1993. The Productivity Paradox - A Clash of Expectations and Statistics. *Communications of the ACM*, December, 1993.

Brynjolfsson E. and Andrew McAfee A., 2011. *Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy.* Lexington: Digital Frontier Press.

Cataldo, M. and Herbsleb, J. D., 2013. Coordination Breakdowns and Their Impact on Development Productivity and Software Failures. *IEEE Transactions on Software Engineering*, 39(3), pp.343-360.

Cheikhi, L., Rafa, E. and Ali, I., 2012. Software Productivity: Harmonization in ISO/IEEE Software Engineering Standards. *Journal of Software*, 7(2), pp.462-470.

Corbin, J. and Strauss, A., 1990. Grounded theory research: procedures, canons and evaluative criteria. *Qualitative Sociology*, 13(1), pp.3-21.

Cricelli, L., Greco, M., and Grimaldi, M., 2014. An overall index of intellectual capital. *Management Research Review*, 37(10), pp.880-901.

Dabbish, L., Stuart, C., Tsay, J. and Herbsleb, J., 2013. Leveraging Transparency, *IEEE Software*, 30(1), p.37.

Dierickx, I. and Cool, K.,1989. Asset stock accumulation and sustainability of competitive advantage. Man-agement Science, 35(12), pp.1504-1511.

Dutta, A. and Roy, R., 2005. Offshore outsourcing: counteracting forces and their dynamic effects. In: HICSS '05, *Proceedings 38th Annual Hawaii International Conference on System Sciences*, Hawaii, pp.1-9.

Edvinsson, L. and Malone, M.S., 1997. Intellectual Capital: Realizing Your Company's True Value by Finding Its Hidden Brainpower. New York: HarperBusiness.

Espinosa, J., Slaughter, S., Kraut, R. and Herbsleb, J., 2007. Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. *Organization Science*, 18(4), pp.613-630.

Fontaine, R., 2007. Cross-cultural management: six perspectives. *Cross Cultural Management International Journal*, 14(2), pp.125-135.

Frey, C. B., and Osborne, M. A., 2014. *The Future of Employment: How Susceptible Are Jobs to Computerisation?* A conference paper in University of Lund, Sweden. Oxford Martin Programme on Technology and Employment. [online] Available at: http://www.oxfordmartin.ox.ac.uk/downloads/academic/future-of-employment.pdf [Accessed 19 October 2016].

Fuggetta, A. and Di Nitto, E., 2014. Software Process, In: FOSE 2014, *Proceedings of the on Future of Soft-ware Engineering*. Hyderabad, India, May 31 – June 7, 2014.

Gorgievski, M.J., van der Voordt, T.J.M., van Herpen, S.G.A. and van Akkeren, S., 2010. After the fire – new ways of working in an academic setting. *Facilities*, 28(3/4), pp.206-24.

Hameed, T. and Nisar, M., 2004. Agile Methods handling Offshore Software Development Issues. In: INMTC 2004, *Proceedings of INMIC 2004*, IEEE Instrumentation and Measurement Technology Conference, Pakistan.

Han, K., Chang, Y. B. and Hahn, J., 2011. Information Technology Spillover and Productivity: The Role of Information Technology Intensity and Competition. *Journal of Management Information Systems*, 28(1), pp.115-145.

Herbsleb, J., Moitra, D., 2001. Global Software Development. *IEEE Software*, 18(2), pp.16-20.

Herbsleb, J. D. and Mockus, A., 2003. An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Transactions on software Engineering*, 29(6), pp.481-494.

Herbsleb, J., Kastner, C. and Bogart, C., 2016. Intelligently Transparent Software Ecosystems. *IEEE Soft-ware*, 33(1), p.89-96.

Hevner, A.R., March, S.T., Park, J. and Ram, S., 2004. Design Science in Information Systems Research. *MIS Quarterly*, 28(1), pp.75-105.

Hofstede, G., 2001. *Culture's consequence: International differences in work related values, revised edition*. Sage Publications, Thousand Oaks, CA.

House, R., Hanges, P., Javidan, M., Dorfman, P. and Gupta, V., 2004. *Culture, leadership, and organizations: The GLOBE study of 62 societies*, Sage Publications, Thousand Oaks, CA.

Huang, C. C., Luther, R. and Tayles, M., 2007. An evidence-based taxonomy of intellectual capital. *Journal of Intellectual Capital*, 8(3), pp.386-408.

Hudson, V.F., 2007. The human touch: cohesive cross-cultural teams begin with savvy relationship management. *Industrial Engineering*, September. Institute of Industrial Engineers, Inc.

Huskins, M., Kaplan, J. and Krishnakanthan, K., 2013. *Enhancing the efficiency and effectiveness of application development*. McKinsey.

Jacob, N., 2005. Cross-cultural investigations: emerging concepts, *Journal of Organizational Change Management*, 18(5), pp.514-528.

Jacobsen, K., Hofman-Bang, P. and Nordby, R. Jr., 2005. The IC rating KM. *Journal of Intellectual Capital*, 6(4) No. 4, pp. 570-587.

Johnson, J., Mawson, A., and Plum, K., 2015. *Effectively Measuring Knowledge Worker Productivity*, White Paper. Iowa: Allsteel Inc., [online] Available at: http://www.allsteeloffice.com/SynergyDocuments/WPA_Insight_WorkerProductivity.pdf [Accessed 16 October 2016].

Jääskeläinen, A. and Laihonen, H., 2013. Overcoming the specific performance measurement challenges of knowledge intensive organizations. *International Journal of Productivity and Performance Management*, 62(4), pp.350-363.

Kamaja, P., 2012. *Role of Intellectual Capital in Company Growth - A Case Study on Technology Companies in Finland*, Ph. D., Tampere University of Technology.

Kamaja, P., Ruohonen, M. and Ingalsuo, T., 2015. Challenges in the Intellectual Capital Evaluation for Dynamic Distributed Software Development Teams – DD-SCALE Program in Progress. In: ECIC2015, Cegarra Navarro, J. G. ed., 2015. *Proceedings of the 7th European Conference on Intellectual Capital*, Cartagena, Spain, 9-10 April, 2015, pp.173-180.

Kaplan, R.S. and Norton, D., 2004. *Strategy Maps – converting intangible assets into tangible outcomes*. Boston: Harvard Business School Press.

Klein, A., 2008. *Organizational culture as a source of competitive advantage*. In: E-Leader, Bangkok. [online] Available at: http://www.g-casa.com/PDF/Bangkok%202008/Klein,%20Bangkok,%202008.pdf. [Accessed 19 October 2016].

Klein, H.K. and Myers, M.D., 1999. A set of principles for conducting and evaluating interpretive field studies in information systems, *MIS Quarterly*, 23(1), pp.67-93.

Kogut, B. and Zandler, U., 1992. Knowledge of the Firm, Combinative Capabilities, and Replication of Technology, *Organization Science*, 3(3), pp.383-397.

Kojima S., Kojima M., 2007. Making IT Offshoring Work for the Japanese Industries. In: Meyer, B. and Joseph, M. eds., 2007. *SEAFOOD '07 1st Conference on Software Engineering Approaches for Offshore and Out-sourced Development*, Switzerland, 4716, pp.67-82. Springer.

Kotlarsky, J., Oshri, I., Hillegersberg, J. van and Kumar, K., 2007. Globally distributed component based software development: an exploratory study of knowledge management and work division. *Journal of Information Technology*, 22(2), pp.161-173.

Lacity, M.C. and Willcocks, L., 2001. *Global information technology outsourcing: in search of business ad-vantage*. John Wiley & Sons.

Lacity, M., Willcocks, L., Cullen, S. (2008) *Global IT Outsourcing: 21st Century Search for Business Advantage*. Chichester: John Wiley & Sons.

Leitner, K-H. and Warden, C., 2004. Managing and reporting knowledge-based resources and processes in research organisations. *Management Accounting Research,* 15(1), pp.33-51.

Lerro, A., Linzalone, R. and Schiuma,G., 2014. Managing intellectual capital dimensions for organizational value creation. Journal of Intellectual Capital, 15(3), pp.350-361.

Löytty K., 2016. *Dynamic distributed knowledge work – Collaboration and productivity factors in the context of global software development*. M.Sc. University of Tampere. Available at: http://urn.fi/URN:NBN:fi:uta-201609022210 [Accessed 19 October 2016].

Löytty K., and Ingalsuo T., 2015. Factors Impacting Successful Collaboration and Productivity of Distributed Software Development Teams. In: Öörni, A., Iivari, N., Kuutti, K., Oinas-Kukkonen, H. and Rajanen, M. eds., 2015. *Proceedings of the 38th Information Systems Research Seminar in Scandinavia* (IRIS 38), Oulu, Finland, August 9-12, 2015.

Marlow, J., Dabbish, L. and Herbsleb, J., 2013. Impression formation in online peer production: activity traces and personal profiles in GitHub. In: CSCW '13, *Proceedings of the 2013 conference on Computer sup-ported cooperative work*. San Antonio, Texas, USA, February 23–27, 2013.

Nickerson, R.C., Varshney, U. and Muntermann, J., 2013. A method of taxonomy development and its application in information systems. *European Journal of Information Systems*, 22(3), pp.336-359.

Oshri, I., Kotlarsky, J. and Willcocks, L., 2007. Managing Dispersed Expertise in IT Offshore Outsourcing: Lessons from Tata Consultancy Services. *MIS Quarterly Executive*, 6(2), pp.53-65.

Palvalin, M., Vuolle, M., Jääskeläinen, A., Laihonen, H. and Lönnqvist, A., 2014. SmartWoW – Constructing a tool for knowledge work performance analysis, *International Journal of Productivity and Performance Management*, 64(4), pp.479-498.

Parmenter, D., 2007. *Key performance indicators: developing, implementing, and using winning KPIs.* John Wiley & Sons.

Prikladnicki, R. and Audy, J.L.N., 2012. Managing Global Software Engineering: A Comparative Analysis of Offshore Outsourcing and the Internal Offshoring of Software Development. *Information Systems Management*, 29(3), pp.216-232.

Reddy, A. and Ryman, A., 2009. *Software development and delivery performance measurement and management: Optimizing business value in software*, [online] Available at: https://jazz.net/library/article/432/. IBM. [Accessed 19 October 2016].

Reifer, D. ed., 2007. *Software management*. 7th ed. Wiley-IEEE Computer Society Press.

Rottman, J. W. and Lacity, M., 2006. Proven Practices for Effectively Offshoring IT Work, *MIT Sloan Management Review*, Spring 2006, [online] Available at: http://sloanreview.mit.edu/article/proven-practices-for-effectively-offshoring-it-work/ [Accessed 19 October 2016].

Ruohonen, M., Mäkipää, M. and Kamaja, P., 2014. Competencies and work practices for dynamic distributed software development in global value networks. In: Passey, D. and Tatnall, A. eds., 2014. *Key Competencies in ICT and Informatics*. Implications and Issues for Educational Professionals and Management, Springer, pp 42-51.

Sahar, F., Raza S. T. and Nasir M. N., 2013. Communication Tools in Offshore Development with Scrum. *IACSIT International Journal of Engineering and Technology*, 5(4), pp. 513-517.

Saxena, KBC., Ruohonen, M. and Bharadwaj, S.S., 2010. Strategic Outsourcing of Engineering Processes Using Agile Methods. In: *Proceedings of the 8th CISTM Annual Conference on Information Science, Technology & Management*, Tampere, Finland, August 9-11, 2010.

Schein, E., 2010. *Organizational Culture and Leadership*. 4th ed. San Francisco: Jossey-Bass.

Schreier, M., 2014. Qualitative Content Analysis. In: Flick, U., ed. 2014. *The SAGE Handbook of Qualitative Data Analysis*. SAGE Publications Ltd.

Ståhle, P. and Grönroos, M., 2000. *Dynamic Intellectual Capital*. WSOY.

Sveiby, K.E., 1997. The Intangible Assets Monitor. *Journal of Human Resource Costing & Accounting*, 2(1), pp.73-97.

Tangen, S., 2005. Demystifying productivity and performance. *International Journal of Productivity and Performance Management*, 54(1), pp.34-46.

The Open Group, 2009. *The Open Group Architecture Framework (TOGAF).* Version 9. ISBN: 978-90-8753-230-7. [online] Available at: https://www.opengroup.org/togaf/ [Accessed 30 September 2016].

Trompenaars, F. and Hampden-Turner, C., 1998. *Riding the Waves of Culture: Understanding Cultural Diversity in Global Business*. New York: McGraw-Hill.

Xiao, M., Nembhard, D. A. and Dai C., 2012. A Survey of Knowledge Work Productivity Metrics, *International Journal of Productivity Management and Assessment Technologies*, 1(2), pp.1-18.

Ziauddin, K. Z., Shahid K. T. and Shahrukh K.Z., 2012. An Effort Estimation Model for Agile Software Development. *Advances in Computer Science and its Applications (ACSA),* 2(1), pp.314-324.