International Journal of Human Capital and Information Technology Professionals

April-June 2012, Vol. 3, No. 2

Table of Contents

Research Articles

Social Capital as a Determinant Factor of Software Development Productivity:
An Empirical Study Using Structural Equation Modeling

Murat Yılmaz, Lero (The Irish Software Engineering Research Centre), Ireland

Rory O'Connor, Lero (The Irish Software Engineering Research Centre), Ireland

ABSTRACT

Social capital is an important network based intangible asset with a potential for maximizing individual and team productivity in a social setting like software development. It is important to investigate intervening factors that challenge software development productivity. In this paper, the authors mixed method approach harnesses a structural equation model (SEM) for its quantitative part to establish a paradigm for understanding the effects of social factors for software development organizations. The proposed SEM model measures the correlations between several potential factors associated with productivity, social productivity, and social capital that are chosen as latent variables. For the qualitative phase, an industrial focus group is used to single out these factors and their association with potential social aspects. Quantitative data is gathered from a survey conducted at a university. The qualitative phase encompasses an industrial focus group, initially starting with the factors from the literature and refined through participants' field experience. Findings indicate that a high correlation exists between several social factors that are reported by the focus group. Finally, initial results suggest that understanding the factors that affect social capital in software development is essential for building and sustaining highly productive development environments.

Keywords: *Social Capital of Software Development, Software Development Productivity, Software Productivity Factors, Software Productivity Improvement, Structural Equation Modeling*

INTRODUCTION

Social interactions and relationships matter for the well-being of a software development organization. These efforts work as a vehicle to convey the diffusion of information during the software development process. Therefore, software development is considered to be a human endeavor (i.e., intellectually intensive team effort) (Fairley, 2009). Consequently, the human and social aspect of software engineering has turned into an important topic to investigate for both scholars and practitioners who strive

DOI: 10.4018/jhcitp.2012040104

to improve organizational efficiency. Today, it is commonly accepted as the productivity of a software development team not only depends on the degree of its members' experience, skills, and competences but also how well its members socially interact. In fact, it is not surprising to observe that experiencing greater production success heavily relies on how the teams socially communicate, and utilize their interactions. These interactions however, should be governed and coordinated to achieve the desired productivity levels both for an individual and a team. By understanding software development as a social activity (Dittrich, Floyd, & Klischewski, 2002), we begin to investigate social capital as a network-based shared norm or a value that supports collective outcomes and to explore the related factors for software development. The goal is to measure the socioeconomic constraints for organizations by exploring several social and productivity factors that are highlighted by software development participants. A software development organization should enable their teams to use their social capital (for example while creating optimal team configurations) and use this value to understand and measure the needs to improve its economically effective levels.

The process of forming a conceptual definition of social capital and customizing a method for its measurement are both very challenging tasks especially in applied settings. Nevertheless, we claim that social capital should be considered as an adequate value to be identified, measured and used for productivity improvement in a software development environment. There are several context specific definitions of the term social capital emerged and used within the empirical research. Social capital should be a factor to improve the productivity of social structure. It should be defined as a network based (hidden) resource identified by the size and the value of an individual's social connections. Therefore, understanding the notion of social capital has a potential of improving social aspects of software development.

Our research agenda has two important objectives: First determining factors that are affecting productivity and second, investigating a method or a model to determine a way to measure the correlations among these factors. In our preliminary study (Yilmaz & O'Connor, 2011), we have analyzed the relationship between the social factors potentially affecting productivity and productivity based upon software productivity literature refined by our focus group studies. Consequently, we designed a SEM model to investigate productivity factors and sought empirical support for our proposed approach. Based on the selected data collected from a preliminary study, in this work, first we plan to revise our previous models, and second we design a new SEM model to examine the relationship between social productivity and social capital.

The rest of this paper is organized as follows: In the next section, we introduce several definitions for the social and value dynamics of software development landscapes such as software artifacts, productivity, the value, social capital, social productivity. The following section describes some of the techniques that are used in social network analysis (SNA). The next section describes the analysis models and methods proposed and used in this research. It presents some of our findings that validates our proposed model, and verifies our empirical approach. Finally, the last section concludes the paper with a brief summary of contributions and some directions for future research.

SOCIAL AND VALUE DYNAMICS

Social dynamics, also known as the dynamics of human interactions, is a multi-disciplinary field of science that is concerned with analyzing socialites or social systems formed by participants and their interactions. This section surveys several important concepts and definitions and the foundations of social and the value dynamics of a software organization. These concepts and definitions highlight the important points of the Social Aspects of Software Engineering (SASE) (Dittrich et al., 2002). Ultimately, SASE helps us to understand social and value dynamics of a software organization. It promotes cooperation within software teams and organizations and helps them to respond better for the dynamic and future trends of software development. We start this section by defining a software artifact. Next, we identify sources of capital that are used in any production process. Moreover, we define both social and human capital, and further the concept of social productivity is introduced.

The Software Artifact

The cost of quality attributes in the software development activities is heavily based on interaction skills of individuals and teams. Specifically, one of the most important of the output of these skills is the software artifact. Software artifact is defined as; "The products, process and software developed by human efforts are considered as artifacts (or tools) that embody human knowledge" (Shariq, 1998, p. 11). "A software artifact is a social artifact, which means it is a product of social beings or an outcome of their behaviors. Software artifact is a unit of material, in the form of a document, presentation, or code, that is developed as a part of, or as a contribution to, the final solution to the users" (Tsui, 2004, p. 3).

According to Baldwin and Clark, an artifact is a *quintessential* outcome of both human intelligence and endeavor. Nevertheless, knowledge-based artifacts (e.g., software, computers, etc.) are interconnected group of entities usually created by a team of workers (Baldwin & Clark, 2000). Morisio et al. point out that the artifacts produced in a software process are complex creations and channeling of the human acumen identified several different characteristics.

Tsui (2004) describes the notion of software artifacts as a "unit of material", which can be in any form such as documentation or source code. Its life-cycle starts from requirements analysis phase follows through product development and documentation. Several entities can be accepted as software artifacts. For example, manuals or guide booklets, and even internal deliverables inside the organization. Software artifacts are considered as smaller and manageable parts of a software project. They are useful touchstones for implementing the concept of separation of concerns (Parnas & Clements, 1986), which values the division of the effort and knowledge by coordinating the software engineering tasks and decisions (Baldwin & Clark, 2000).

Shariq suggested that the knowledge should be considered as an outcome of human activities, which essentially produces knowledge artifacts, and knowledge networks are intervened by these artifacts (Shariq, 1998). Cluts conducted a case study to develop a framework based on the connections between people and their activities where artifacts are described to contain a backlog of the past events and connections among them (Cluts, 2003).

Productivity

In so far, as it is not different from other forms of industrial production, software production is considered as an economic process of conversion of inputs to outputs based on industrial methods of manufacturing. Consequently, one of the concerns of software process improvement is investigating methods to measure the factors affecting software productivity. However, software development productivity depends on a combination of several related factors. Therefore, it is very challenging to develop a method that efficiently obtains the correlations among the factors affecting software productivity.

In general, economic productivity is considered as a value to measure the efficiency of this production process. For example; it is measured as a ratio of the units of inputs versus the units of the outputs (Misterek, Dooley, & Anderson, 1992). However, it is also considered as a utilization of resources with an optimal cost (Prokopenko, 1987) (i.e., a ratio of production capacity to production cost). Productivity apparently depends on the availability of resources and highly connected with the value creation processes (Tangen, 2005). Brynjolfsson (1993) states that it is an essential economic criterion for the contribution of any technology to an economy. Sink et al. (1989) defines productivity as a ratio between the actual outputs versus the expected resources that has been used. Based on the assumption that time is a resource; Jackson and Petersson (1999) suggest a time-based measurement of productivity (i.e., a ratio between value adding time versus the total time). The limitation of this approach is that usually there could be a lack of information about the resources that are consumed during the production process.

Similar to several other industrial propositions, software productivity is traditionally defined as a ratio between the inputs (e.g., the cost of work/resources) versus the outputs (i.e., software artifacts or services) within the production process of software development (Chemuturi, 2009). However, empirical evidence suggests that it is hard to find a suitable way for measuring productivity (Tangen, 2002) in industrial production and software development productivity in particular (Jones, 2009). Because, it may be considered differently for stakeholders from their distinctive perspectives. For example: from developers' viewpoint, a productivity measure would be the amount of code produced for the software system, on the other hand from the users' perspective; it could be the degree of functionality achieved for the software system.

According to Jones, teams that are building similar kind of artifacts can easily progressed to more mature stages on software productivity because this improves the experience levels of both managers and software teams (Jones, 2007). However, depending on the quality and maturity levels of a software artifact (e.g., source code, test materials, documentation), he argues that reuse could have not only a positive impact but may also a negative affect both for deliverables as well as productivity improvement efforts as a whole. For example, using high quality reusable artifacts could improve productivity; however this can also reduce the productivity of a software project because such an artifact may not be near zero defect levels.

Productivity is significantly affected by the quality of workforce, management capabilities and environmental conditions of a software organization (Selby, 2007). Moreover, the effective usage of methods and processes, project complexity, software team morale, and effective team configurations are the key adjustment factors for software development productivity (Jones, 2007). However, interdependent factors involve with productivity can't easily be controlled or improved by only manipulating the variables such as dynamic motivational factors, cost of communication and social expenses (Abdel-Hamid, 1996).

Several software engineering researchers suggest methods to improve software productivity by balancing the field of tension among the people (regarding to their activities), processes (with respect to its tasks) or technology (by its advances in computing power) (Gilb & Finzi, 1988; Scacchi, 1995; Hantos & Gisbert, 2000). There are several attempts in the literature that aims to measure software productivity. For example, Scacchi suggests a framework for examining and measuring software productivity to perform a simulation over the production dynamics of software projects (Scacchi, 1995).

One common approach for improving software productivity relies on the theory of group productivity by psychologist Ivan Steiner (1972), who states that consequences of defective processes are important for explaining actual productivity (Hamid & Madnick, 1989). It is calculated by subtracting these defects from potential productivity ("*Actual Productivity = Potential Productivity - Losses Due to a Faulty Process*"). Hamid and Madnick (1989) also state that potential productivity is explained as; if an individual or a group uses the maximum potential of its resources than a level of maximum productivity is achieved. They added two factors are important for representing the shortfalls for software quality and productivity problems: (i) the task's characteristics (i.e., complex nature of a task), and (ii) team resources (i.e., fitting individuals or team skills over tasks and tools). These factors could increase the cost of communication and lower the motivation of individuals and software teams.

A common view in engineering terms is that the productivity improvement indicates producing more outputs from a known set of inputs by reducing the influence of any factor that hinders productivity. For example, software productivity improvements can be achieved by having a skillful team, improve the path of development by reducing rework, and by creating reusable and more manageable software artifacts (Boehm, 1987). In fact, an increase in the productivity is achieved when human resources used in the software development process starts adding more value to the software product.

Over the past few decades, software productivity has been investigated by using several indicators affecting the productivity. Pfleeger (1991) conducted a productivity investigation by using a statistical method called regression analysis. By using this technique, he constructs an estimation model of productivity where he calculates the effects of cost factors in a predictive manner. Moreover, regression analysis has also been applied for determining the correlation between size and effort for software development projects (Maxwell, 2002).

Despite a considerable amount of literature has been published on productivity factors affecting software organization (de Barros Sampaio, Barros, de Aquino, e Silva, & de Lemos Meira, 2010), several questions still can be found unanswered (e.g., a correlation and/or significance among these factors). For example, it is considered as hard to address a single solution to the several issues of productivity identified (Boehm, 1987).

Team size can also be considered as an important factor for understanding the change in development productivity, as well as project size and complexity. Most importantly, creative and talented individuals are the main components of a productive team in software development projects (Curtis, Krasner, & Iscoe, 1988). This suggests that neither of the other factors can produce more significant weakness as a lack in human capital.

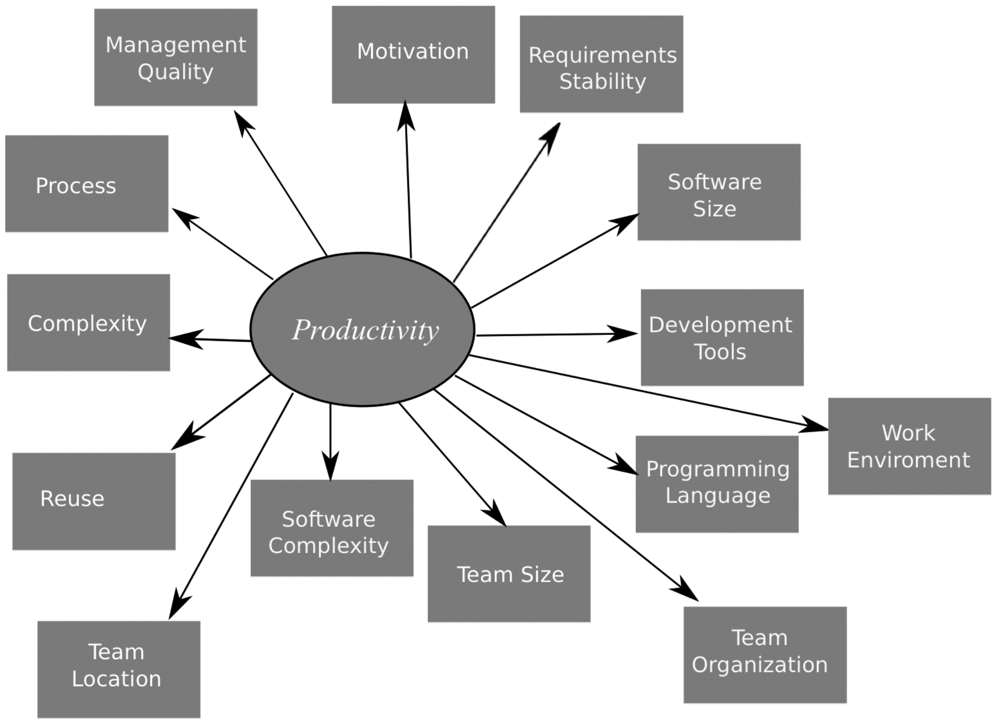
Boehm performed significant amount of software productivity research (Selby, 2007). Moreover, there are several software productivity factors identified in literature: motivation (Boehm, 1987; Beecham, Baddoo, Hall, Robinson, & Sharp, 2008; Sharp, Baddoo, Beecham, Hall, & Robinson, 2009), process (Boehm, 1987) and process design and coordination issues (Chiang &Mookerjee, 2004), complexity or a size of a project (Scacchi, 1995; Boehm, 1987, 1981), reuse of software artifacts (Lim, 2002; Boehm, 2002), communication demands regarding to team size (Scacchi, 1995; Blackburn, Scudder, & Van Wassenhove, 2002). Based on the research surveyed in this paper, Figure 1 shows several productivity factors identified for software development.

The Value

A software development process aims to create an economic value for all the investors in the enterprise. Boehm (2003) claims that many software engineering projects are considered as performed within a value neutral setting. In other words, every task and activities are regarded equally important without considering the outcomes and business value propositions. However, researchers suggest that many reasons (e.g., lack of utilizing project resources, fail to prioritize project requirements) that cause software projects failures should stem from the problems of value-neutral approaches (Biffl et al., 2006).

According to the software engineering perspective, value creation activities mostly focus on the economic significances, e.g., customers' requirements and the things stakeholders are valuing the most (Selby, 2007). However, the stakeholders' contributing the value creation process has different considerations and therefore different goals and expectation from the same software system. Furthermore, they might have subjective definitions of the value. Halling et al. (2004) considers the relationship between value and the project attainment and defines the goal of a project is to produce greater value than the values of the resources consumed by

Figure 1. A productivity model based on factors affecting software development (adapted from de Barros Sampaio et al., 2010)



the investment in the software organization. Boehm and Sullivan (2000) suggested that the best way to establish the uttermost value from software project resources by administering the software development process as an economic activity of investment. The knowledge, which is accumulated as the capacity of an economic activity, is ultimately based on the human capital, which also directly influences the efficient use of physical capital (Welfe, 2009).

Human and Social Capital

The classical notion of capital states that the capital becomes apparent from the social interactions between capitalists and laborers. In other worlds, it is an end product of a social process. According to Marx (1889), it is a surplus value captured by individuals who control the production processes. In addition, it is also a kind of activity of investment for the resources so as to gain profit. Capital is not only a result of the process of manufacturing but also an outcome of trading products and goods based on the social relations between capitalists and laborers (Marx, 1889).

In the last decade, this classical viewpoint evolved to include the intangible assets for human intensive organizations such as the economic value generated by human and social capital. Understanding and measuring human capital is a challenging process; evidence suggests that quality of social and organizational relations based on several individuals' interaction affects the sustainability of any social structure. Human capital theory relies on the fact that, laborers become capitalists by accumulation of knowledge and skills and therefore human experiences are embedded inside the notion of capital (Welfe, 2009). It simply states that the laborers are trained in specific subjects and captured valuable experiences in their work life somehow become irreplaceable through the production processes, which also constitutes competitive advantage for a software development organization. One form of human capital encompasses several intangible assets such as the personal social network of resources of an individual is also known as social capital.

Social Capital can be defined as an intangible resource, which benefits from social connections and networking. It may include the opportunities that an employee's social network can provide. Lin defines social capital as "investment of social relations with expected returns in the market place" (Lin, 2002). Bourdieu understands the term as a presentation of actual and future resources that are linked as a network of relationships (Bordieu, 1986). His definition designates that social capital is based on two components; social relationships which affords possibilities to help them obtaining accessibility to the resources by their relationships, and resource quality. He claims that the value of social capital, which is based on social connections, should easily be convertible to an economic form of capital.

There are other definitions of social capital (Portes, 1998). For example, (i) it is considered as a resource that individuals yield from social structures regarding to quality of their relationships, (ii) it is an observable pattern in a social structure which influences the relationships among the individuals or social groups, and (iii) it is the quality of personal contacts which individuals gain to increase both financial and intellectual capabilities (Bordieu, 1986; Portes, 1998; Lin, 2002). Fukuyama defines the social capital as "the ability of people to work together for common purposes in groups and organizations" (Fukuyama, 1996, p. 10). Later in his works he considered the term as an intangible value obtained from social groups that promotes collective outcomes. He argues that social capital is dependent on norms like honestly, trust and dependability.

Social capital comprises resources to be captured by individuals (Coleman, 1994). According to Portes (1998), social capital is inherent in the fabric of actors and relationships. In order to own a social capital one should have linked with others, therefore, it should be measured with quantity of social connections that an individual might have (Portes, 1998). Coleman concludes that all kind of social structures henceforth relations enable some form of social capital. In fact, the individuals intentionally connect with one and other to form social networks and expect benefits from these actions (Coleman, 1988). Here, we define the term as;

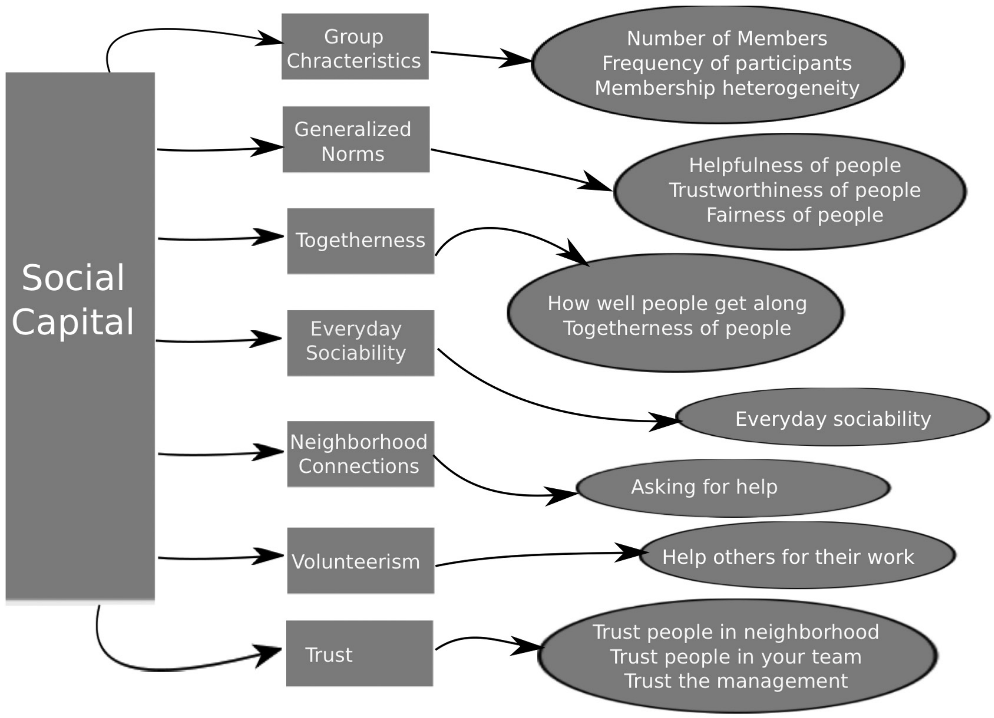
Social capital *is a potential form of intangible resources based on patterns of social connections and social abilities of individuals, teams or social groups that has a potential to contribute to the economic progress of an organization.*

The higher level of social capital attainable by participants of a software development organization should help to improve the productivity of teams and individuals in a software firm. Consequently, leveraging the social connectivity in a software development organization should have positive impact on the productivity of a software development group. Aligned with the improvement efforts, this can be considered as one of the actual benefits of social capital obtained from networks of relations. Exploring and implementing team based social improvements should help us to improve structural and organizational stability. It should, therefore, enable us to constitute more cohesive information exchange networks, which may have a positive effect on the productivity of a software team. An illustration for dimensions of social capital (Narayan & Cassidy, 2001) is shown in Figure 2.

Social Productivity

Humans are social creatures. This means they usually depend on others and prefer to live in interacting groups (or socialites) where they influence one another. In fact, they continue to be increasingly interested in establishing a society and improving social outputs of their organized groups. Thenceforth, they prefer work in teams

Figure 2. Dimensions of social capital (adapted from Narayan & Cassidy, 2001)



and inclined in order to form more complex outputs. By considering the social behavior itself as a method to exchange of goods (Homans, 1958), they create and share knowledge-based outcomes (in forms of artifacts), and have their experience pass through further generations so as to improve the economic well-being of a society. The notion of a society should be considered as the end product of individuals' decisions and social interactions. It is therefore not surprising to discover that there is a visible relation between a concrete social structure and the mental structure of a socio-economic group (Bourdieu & Wacquant, 1992).

For many researchers productivity is an economic concept, however, it also has a sociological aspect which is highlighted by Barnett "While an economic concept of productivity is undeniably important in explaining the material wealth of groups, personal observation suggests that understanding organized groups-including business firms-requires a sociological concept of productivity" (Barnett, 2004, p. 739).

Moreover, Barnett claims that social productivity happens when a team or group of people interacts and create social interactions and outcomes, which certainly effects the functioning of teams (Barnett, 2004). It should also portray the actions and reactions of a social organization. The economic perspective suggests that individual's actions are established, directed and limited by interpersonal trust, social networks and organizations (Coleman, 1988). These interactions are based on group needs, values and action of the group and further shapes new actions or action sets. Consequently, Barnett (2004) indicates that there are mainly four social constructs (outputs), which matters and varies among organized groups namely: (i) reputation, (ii) symbols, (iii) trust, and (iv) perceptions of fairness. These outputs promote ties around a social group and usually help individuals to work for the benefit of a team as a whole (Barnett, 2004). All of these outputs are mainly caused by social interactions and have an impact on the productivity level of a social group.

Therefore, we argue that social productivity can be a vital component for understanding the structural complexity of a society. All constructs defined above can be considered as resources of a group ("ingredients of social capital"), as well as the outcome of functioning of a group ("features of social productivity") (Barnett, 2004). Here, we define the term as;

Social productivity *is an intangible asset as we termed here to reify the effects of social factors on the social and economic landscape of a software organization. Therefore, a new kind of productivity improvement should be considered as the transformation of social capital (potential energy) into social productivity (kinetic energy) form.*

In other words, social productivity represents an identified stock of social capital that is transformable to value creation activities so as to form software artifacts. The notion of social productivity seems useful for achieving software productivity improvement goals. From a socio-economic viewpoint, it investigates ways to improve collective outputs, which enables a software development organization to make economic progress. These organizations build on the idea of collaborative social activities, which could be an identifiable component of teams that work in the favor of software organization.

In the socio-economic landscape of software organizations, social productivity should represent a concept for advancing the ability of software development organizations by understanding the factors that hinder social development and structure. It is, therefore, important to seek ways for increasing the efficiency and productivity of individuals, which depends on the subset of various factors mentioned below such as quality of their social interactions, and communication effectiveness of its members for their contributions to collective outputs, etc.

A model based on the factors affecting social productivity of software development is illustrated in Figure 3. This paper recognized that the potential factors affect social productivity of software development are: (i) the reputation of a team leader or a product leader (Stober & Hansmann, 2009), (ii) communication as the level of social interactions among the team members (Dittrich et al., 2002), (iii) information awareness as knowing what other teams and individuals doing (Koh & Maguire, 2009), (iv) trust, for collective results on working environment (Anderson, 2004; Hazzan & Dubinsky, 2008), (v) a social life, i.e., socialization with teammates (Kelly, 2008), (vi) fairness (Hazzan & Dubinsky, 2008), and (vii) frequent meetings (Churchville, 2008).

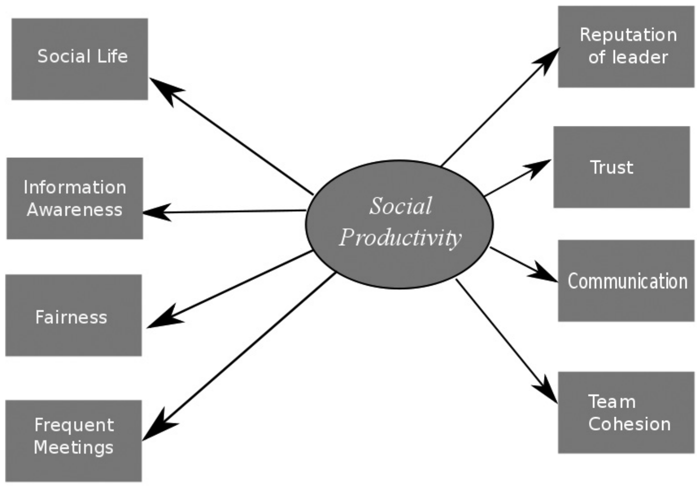
SOCIAL NETWORKS

This section primarily focuses on the notion of social networks, which is useful for understanding people's network of social interactions, especially in knowledge-driven organizations. From the outset, there are two interesting aspects of social networking which seems relevant for understanding the dynamics of team structure in the context of software engineering organizations. First it should help to observe the diffusion of information through the structure of software development organizations. Second it may reveal dynamic changes in interrelationships and their impact on the economic outcomes as a whole.

In particular, it is important to explain relationships of individuals, which constitutes the social structure of an organization. To this end, one may observe the flow of information among the individuals and teams. Social scientists claim that identifying (micro level) changes in social units those affect macro level organizational mechanisms may have a significant impact on the business processes.

Social interactions constitute a basic connectivity process, which enable circulation of

Figure 3. A social productivity model based on factors affecting software development



the flow of ideas, and decisions throughout the organization. This process is composed of connections that are the driving force for community creation processes, which create a network of social interactions called social networks. They are virtual ties (structures), which are able to represent many of our social interactions (e.g., friendships, organizational) among the knowledge processing sociable units (Bavelas, 1950). These units may contain several types of actors (e.g., teams, individuals, groups) somehow attached by some special associations (Jackson, 2007).

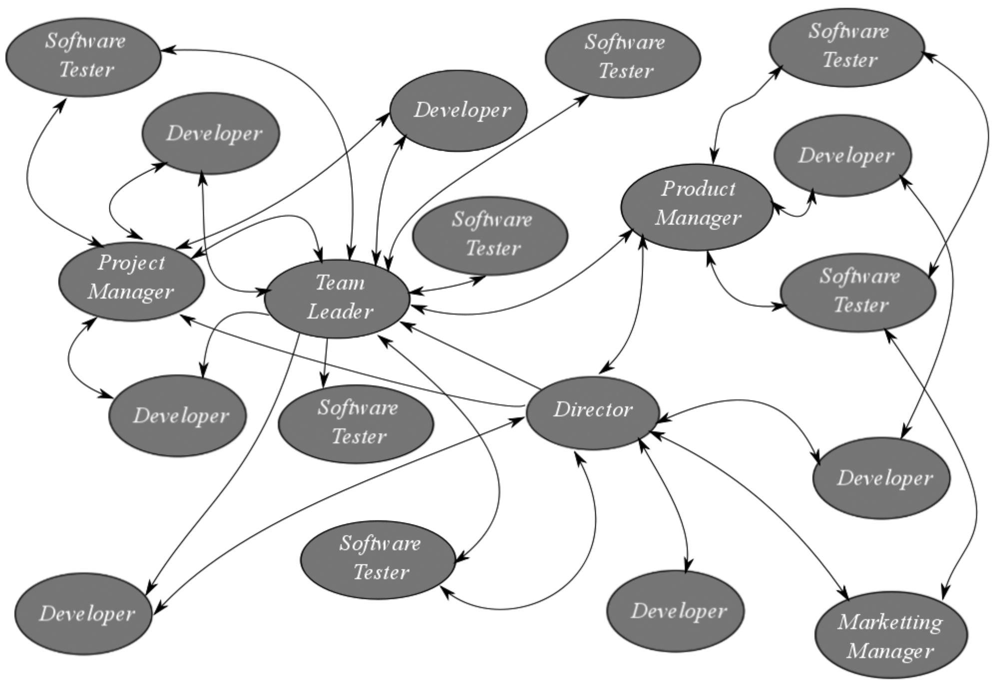
While sociologist have been working on the social networks for a very long time, today, and several other disciplines are inclined to identify these kind of networks in their formal studies (Barabasi &Frangos, 2003). The features of social interactions, the information flow, or the entire structure of a software company can be visualized for various reasons. For example, to establish a network for understanding organizational structure, to observe team formations, to transport social influence, and to increase the quality of information sharing among the organization, etc.

Social Network Analysis

SNA is a study of measuring and understanding the information flow in any social organization or an environment. Frequently, it is used for visualizing ties, relations, performance and interactions among the individuals, teams or an entire organization. This interpretation suggests that SNA is a good way of estimating relationships such as deep structural patterns of a software team or an organization (see Figure 4 for an example). From this viewpoint, it is not only useful for mapping individuals and their relationships with one and other but also important for identifying the connection and communication problems. A recent study by Kratzer et al. (2009) found that SNA might be a competent managerial instrument for assessing the interactions emphasized among knowledge-intensive teams. Unlike organizational charts, which show formal hierarchal connections, SNA can concentrate on all types of relations between actors. Therefore, it should be helpful for efficiently investigating networks of people and their connections that constitute the social (working) units embedded in the fabric of an organization (Barabasi & Frangos, 2003).

There are several social network characteristics that are identified and analyzed by

Figure 4. Social network map of a small company



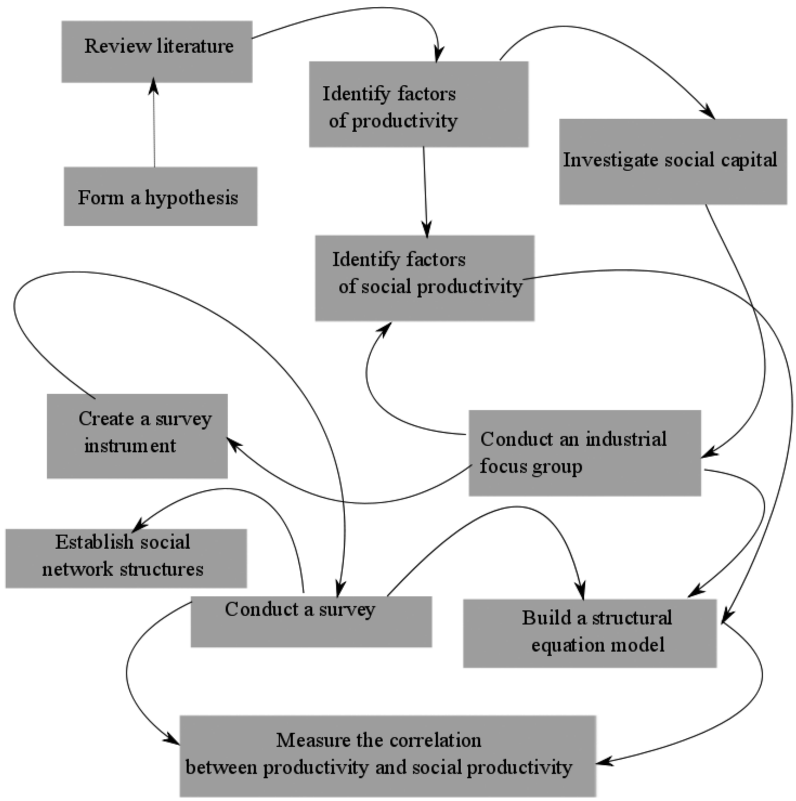
researchers. For example, size of a network is a critical feature of a social network (Wasserman & Galaskiewicz, 1994). It helps for making comparisons among social units and shows the complexity of the structural form. The concept of degree identifies the sum of the connections between actors. Density of a network shows the cohesion and the recursion of the nodes inside the network. Centrality of a social network measures the alignment of actors with respect to center of the structure whereas reachability checks the nodes and the actors whether they are at a reachable point (Wasserman & Galaskiewicz, 1994). Centrality sometimes referred as "social status" of an individual, to whom everyone is willing to be connected.

Lazer (2001) introduced two SNA terms; network elasticity, which defines how private is the network (i.e., can actors choose who they can connect?) and individual plasticity, which examines the impacts of connections over the individuals along the social network. Further he suggested that different social systems are varying with respect to elasticity of their network and plasticity of their actors (nodes) (Lazer, 2001).

Research Model and Methods

In this section, we describe the research model and methods used in this study. In order to investigate social capital and factors that are affecting its existence in software development, we introduce a mixed method technique, which encompasses both qualitative and quantitative phases. We formulate our initial model based on the social capital literature and in our qualitative phase we use an industrial focus group to discuss several factors in software development settings. We conduct a survey to collect data about these factors from a university environment. Furthermore, we use our enhanced model to measure factors affecting social capital and software productivity. In addition, we use a simple method for establishing social network

Figure 5. Updated systematic approach for software productivity research (adapted from Yilmaz & O'Connor, 2011)



of participants by simply asking them about their contacts in a classroom setting.

The Systematic Approach

Here, we develop a systematic approach to address the relationship between productivity and social productivity (Figure 5).

- First, based on a structural equation model, we establish a hypothesis: There is a high correlation between productivity and social aspects that are affecting it such as social capital, social productivity, etc.
- Second, we survey the literature to seek and identify factors that have a negative and a positive impact on productivity
- Third, we conduct a focus group research and consulted a software professionals for their opinion about these identified factors, especially in an industrial setting and consequently utilize this information to change some our initial settings.
- Fourth, we create a survey instrument for testing and validating identified relationships, and several observable and latent variables we proposed. Further, we ask participants about their contacts on environment.
- Finally, to test our structural equation model, we conduct a survey to collect data at a university environment.

Structural Equation Modeling

The social and economic factors affecting productivity rely on people based measurements and observations. Therefore, they are mostly qualitative by their nature and sometimes hard to quantify. It is therefore not surprising to discover that a precise measurement model is hard to construct. For example, a software productivity model is a complex theoretical model, which should rely on a solid (stochastic) statistical approach to address quantification problems. In general, a complex model represents several associations, and causal relationships that should be explicitly testable and justifiable by empirical data.

Frequently used in social science studies, a family of flexible interrelated statistical techniques (i.e., multivariate, multiple regression analysis, factor analysis) for analyzing empirical data and testing variables and evaluating their network of hypothesized relationships is called structural (simultaneous) equation modeling (SEM) (Hayduk, 1987). Based on patterns of statistical expectation, it is a confirmatory multivariate (multi equation) analysis technique for estimating the structural or casual relationship among variable types (observed and latent), and specifying relations among these latent variables both for linear and non-linear structures. SEM models use a collection of simultaneous equations based on a combination of observed and latent variables (hypothetical constructs or factors), which are frequently used by sociology, psychology research and econometric research (Kline, 2010). The main component of a structural equation model is a form an initial hypothesis includes the factors that may be connected, and are assessed by several statistical tests and if necessary adjust through modification indexes.

According to Bollen (1989), SEM has a structure based on three different methods of statistical research: (i) analysis of covariance structures and estimations, (ii) modeling of latent variables (factor analysis), and (iii) path analysis (Blanche, Durrheim, & Painter, 2008). A typical SEM structure has up to three simultaneous equations, which includes: (i) a measurement model that can have dependent variables, (ii) a sub-model with independent variables, and (iii) a structural sub-model for concurrent estimations (Kline, 2010).

The holistic view depicts both type of variables (e.g., dependent, independent), for example, observed variable with weighted averages on unobserved variables where a conditional dependency may occur (Hoyle, 1995). A complete SEM model includes both structural and measurement model (Hoyle & Smith, 1994). In practice, the whole model structure rarely used, therefore there are several SEM models without measurement features (SEM with observable parameters) or only have the features of confirmatory factor analysis, i.e., measurement facilities (Kline, 2010).

To sum up, SEM is a quantitative approach, which provides an assessment of relationships among the interrelated variables. It is a method to provide empirical validation for investigating a hypothesis or a situation, in which there are relations between several dependent and independent variable. In fact, it is a statistical model based on the intensity of relationships between variables, which supports a proposed hypothesis, and simultaneous interrelations (Lovric, 2011).

Although it is a quantitative approach, SEM offers a start from a qualitative viewpoint; it has the ability to show how the chosen factors or variables are not only correlated but also interrelated to one other. Therefore, it can be helpful for observing the relationship among several coefficients. It enables us to investigate how a hypothetical model might be effectively fit with sampled data. In particular, a model based on the combination of regression, path, and confirmatory factor analysis should be useful for analyzing social factors and their interdependencies.

In the next part, we illustrate three SEM models, which aim to visualize productivity and social productivity, and social capital as latent variables and consider several productivity factors from the literature to measure the correlation between productivity and several productivity factors, social productivity and related factors, and finally social capital and associated factors. All of these affecting factors are identified from the literature and chosen by a focus group, we formalize models by using the data provided from the conducted survey.

The Measurement Model

We hypothesize our structural equation models based on a couple of latent variables and some observed variables, where their measurement relies on various identified factors. We initially chose productivity and social productivity (and later social capital) as latent variables for our structure equation relations. We presume that these variables are constructs that are identifiable by using several factors surveyed from the productivity literature.

Although, it may be difficult to obtain rich and insightful data from practitioners in a specific area of interest using both qualitative and quantitative research method, evidence suggest that a focus group can be an efficient way to reach that information (Krueger & Casey, 2009). Focus group setting may be ideal for people to build new ideas on the top of other's opinions and further discussing their experiences. After having chosen factors of both productivity and social productivity, a focus group study was conducted to investigate opinions of software management teams in a middle size software company. The discussion group was composed of nine personnel from the management team and CEO of a software company (total ten participants). The meeting lasted three hours, which was held at a conference room of this company. As suggested by Krueger, the session was facilitated by one of the authors who commenced an introduction to encourage participants and initiate the discussion settings (Krueger & Casey, 2009).

We asked the management team about their opinion on productivity factors and one individual from the management team took written notes. A guide containing five questions and a preliminary model of social productivity was prepared for the focus group discussion:

- What is your definition of productivity in software teams?
- What is your opinion of the factors that are affecting the productivity?
- What do you think of the most important factors among these ones for productivity?
- How would you describe the social factors of productivity?
- What is your opinion of the social factors that are affecting the productivity?

The goal of the focus group study was to identify opinions from industry about the most important factors that are affecting for productivity, social productivity, and social capital. One of the participants defined productivity as working faster, while one other introduced the term efficient to this definition. Participants discussed social aspects of productivity including the impacts of social values over productivity, the communication frequency, coordination efficiency, team augmentation, and task rotation. In addition, the group discussed the selected items from the software productivity literature; the impact of complexity or size of a software project, and re-usability of the created software artifacts. After having a debate on several factors affecting productivity, the group decided by voting that complexity of a project and re-usability of software artifacts are more important than some other factors from the literature. Consequently, focus group activity provides us an opportunity to discuss our opinions about productivity factors based on industrial setting. We refined our list of factors by using the information provided in this session.

Our first and initial model is based on factors affecting productivity and social productivity. By using the productivity literature in general, and software development productivity in particular, our focus group chose five factors that have been mostly referenced by software productivity literature, that are (i) Motivation, (ii) Process, (iii) Reuse, (iv) Complexity, and (v) Team Size. Based on the discussion with our industrial focus group, we decided to use four observed variables as factors affecting social productivity, which are (i) Leadership, (ii) Trust, (iii) Communication, and (iv) Team Cohesion. For the social capital part of the research, we chose (i) communication transparency, (ii) social relations, and (iii) regular meetings at first to identify the factors affecting social capital. Later we add (iv) helpfulness to our model as an additional factor for social capital research.

In light of these discussions, we designed a survey instrument to measure the impact of the identified factors on both productivity and social productivity. We used 5-point Likert scale (i.e., a psychometric scale frequently used in social research) to measure the relations. Furthermore, we add several other questions, in which participants were asked to rank these factors regarding to their priorities and importance in their working environments.

Productivity Factors as Structural Relations

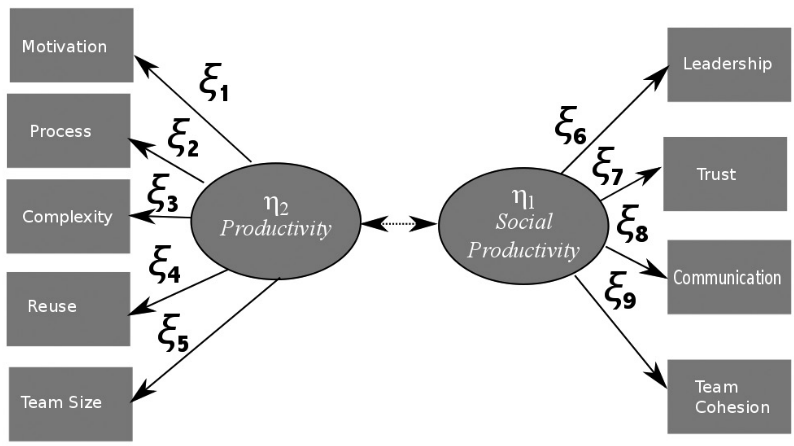
The first model relies on the fact that productivity and social productivity can be presented as latent variables showing themselves through a set of measurable factors or indicators selected from the literature (Yilmaz & O'Connor, 2011). Based on previously identified factors affecting both latent variables, we draw a model of social productivity by using SEM diagramming (Figure 6) for a conceptual representation of the hypothesized model. Traditionally in SEM modeling, observed variables are illustrated in rectangular boxes and the latent variables are shown in circular boxes. Moreover, the lines connecting the variables illustrate the direct effects of the indicators on the latent variables.

First, we proposed social productivity and productivity as latent variables which are observable based on four and five measurable indicators (observed variables) respectively (nine indicators in total). To conduct SEM, we use LISREL - a software package frequently used for structural equation modeling (Joreskog & Sorbom, 2001). The data was collected from surveys obtained from university student. In particular, for this paper, we select postgraduate students and fourth year students (we have around hundred participants in total)

After constructing a hypothetical model (proposed association based on a literature review) and illustrating it on path diagram, there are two steps commonly used in SEM analysis and modeling. First, we explore a measurement model, which evaluates the relationships between measurable indicators and latent variables (constructs). Second, using the results of the measurement model and several statistical tests, we investigate a structural equation model for an acceptable good fit. Following frequently used SEM notation, here, the first hypothesized model is presented in Figure 6, where observed variables are depicted by rectangles and latent variables are illustrated by circles, and further lines are used for portraying the relationships among the variables.

Although there is no actual agreement on which fit index is the most appropriate one (Ping, 2004), several researchers suggest that a chi-square test should be useful for evaluating how well the model fits with the data. This test relies on the idea of measuring the difference between hypothetical model and data by using the concept of null hypothesis or an independence model, which assumes that variables among the relationships are uncorrelated. A high value from a chi-square difference tests shows that there could be a deviation between data and the model. However, the reliability of this test may decrease when sample size reaches a significant volume. Therefore, several type of indexes are simultaneously used for investigating relationships between the model and the data including: Goodness of Fit Index (GFI, a measure to fit model and covariance matrix), adjusted goodness of fit index (AGFI), root mean square error of approximation (RMSEA - how well a model fits the data with respect to degrees of freedom), normed fit index (NFI), and comparative fit index (CFI) for assessing refined model relative to fit an independence model. For GFI and CFI, value above .90 should be acceptable and for RMSEA a reasonable fit could be obtained with a value below .08.

Figure 6. Initial conceptual model for social productivity of software development (Yilmaz & O'Connor, 2011)



In our first model, a null hypothesis is totally rejectable where $\chi2$ (36, N = 204) = 2277.38, p < .001. Consequently, the measurement model (Figure 7) for the collected data is as follows, $\chi2$ (26, N = 204) = 67.33, p < .001, where RMSEA = .088, GF I = .93, AGFI = .88, CFI = .98, NFI = .97), where all of the structural correlations between observed and latent variables were statistically significant (p < .001). A chi-square difference test indicated significant improvement in fit between the independence model and the hypothesized model, $\Delta\chi2$ (10, N = 204) = 2210.1, p < .001). Productivity coefficients are ranged between Motivation (structural coefficient= .86, p < .001) and Team Size (structural coefficient= .71, p < .001). Motivation has strongest correlation with productivity among all other productivity factors, while Communication has the strongest correlation among social productivity indicators.

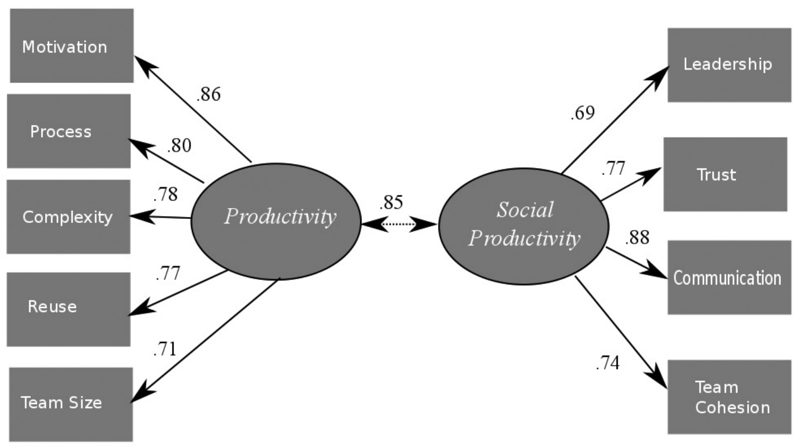
In the next step of our analysis, we hypothesize a structural model, which includes social capital as a latent variable (Figure 8), and we use three indicators affecting the social capital including: (i) communication transparency, (ii) social relations, and (iii) frequency of meetings. The constructed model was tested for a good fit for the data, $\chi2$ (19, N = 204) = 46.24, p < .001), RMSEA = .084, GFI = .95, AGFI = .90, CFI = .99, NFI = .97). Results of a chi-square difference test indicated that a significant improvement in fit, $\Delta\chi2$ (9, N = 204) = 1547.85, p < .001).

Based on the two previous models, it seems reasonable to suggest that social capital has a high correlation with the productivity of a software process. As seen on the path diagram, we initially chose communication transparency, social relations, and regular meetings as the predictors of social capital, respectively with structural coefficients = (.93, .71, .40, p < .001).

Specifically, this paper investigates a structural equation model, which identifies the correlation between social capital and social productivity (Figure 9). It uses both factors of social capital and social productivity, which was chosen by our focus group. Moreover, we use a new factor for social capital, i.e., helpfulness as suggested by our literature survey. This model demonstrated a reasonable fit to the data. The structural path coefficients, and all factor loadings are significant, $\chi2$ (19, N = 204) = 13.44, p < .001), GFI = .98, AGFI = .97, CFI = .99, NFI = .99). Results of a chi-square difference test indicated that a statistically significant improvement, $\Delta\chi2$ (9, N = 204) = 1253.53, p < .001).

The extended model hypothesizes that there is a correlation between social capital and social

Figure 7. Structural equation model for social productivity of software development



productivity. In this model, social capital is identified by helpfulness, communication transparency, social relations, and regular meetings with structural coefficients = (.70, .72, .83, .42p < .001) respectively. Social productivity is identified by leadership, trust, communication, and team cohesion with coefficients = (.75, .80, .82, .77, p < .001).

Social Network Maps in Applied Settings

Social network maps are useful for investigating the importance of connections that particular individuals have established in a social setting. These connections serve the understanding of diffusion of information, which is essential for creating productive team configurations.

Figure 8. Structural equation model for social capital of software development

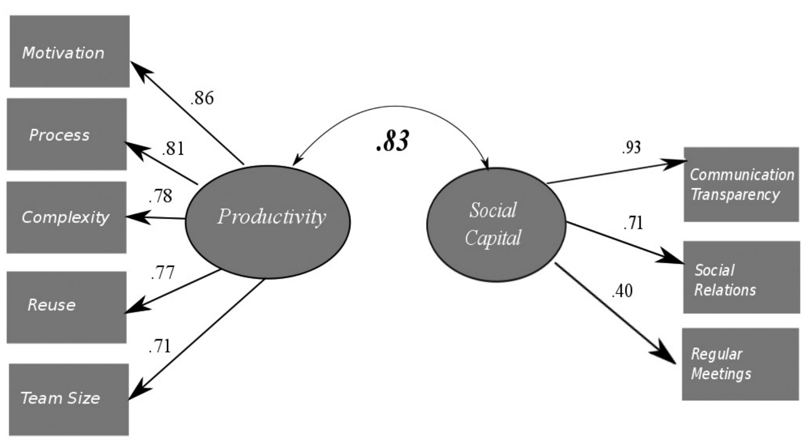
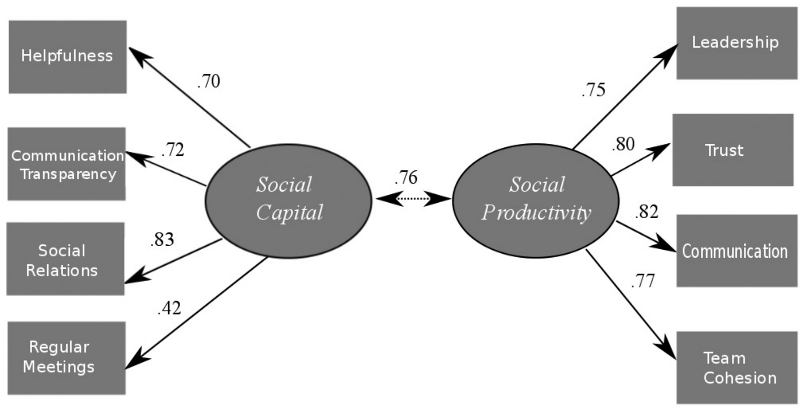


Figure 9. Extended structural model for portraying the relationship between social capital and social productivity of software development



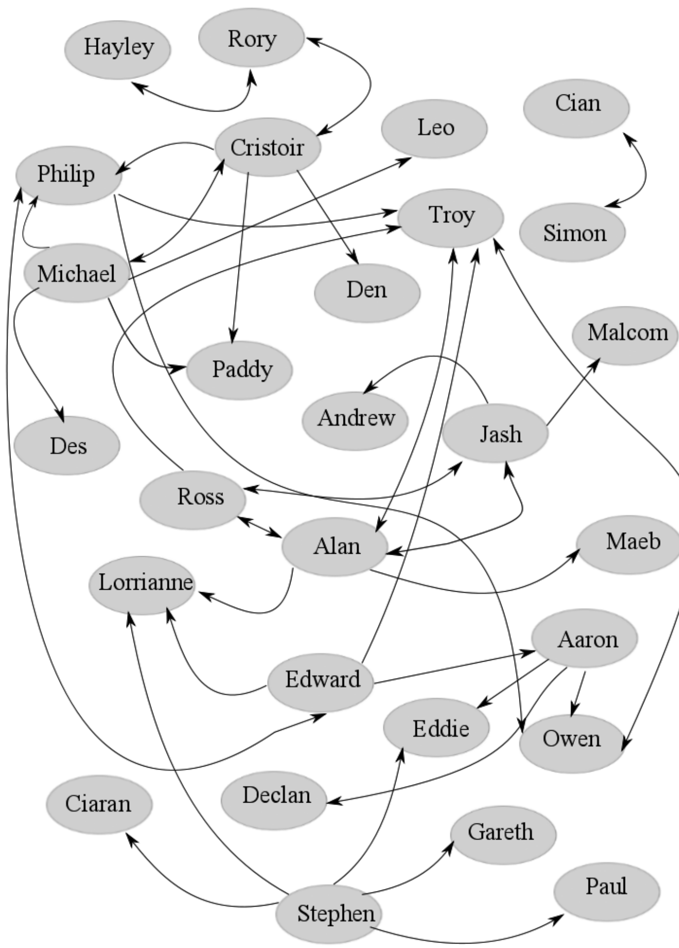
Here, we visualize a part of a social network of participants where we collect our survey data (Figure 10). This illustration suggests that there are key participants in a network constellation based on several connections as expected. According to our survey results, nearly all of the identified key players in a social network think team interaction is the most important factor among the others, and therefore they ranked this indicator as the most important one. Moreover, evidence suggests that degree of centrality (people identified as key players) is vividly observable even within a small social unit of participants. In other words, participants who are voted as key players have chosen as a primary contact by many of the participants, which highlights their importance for the integrity of a social structure. Sportingly, key players and people visualize around them dominantly ranked trust as the most important factor for improving social productivity. Individuals who have more connections (i.e., with a potential of higher social capital) in the environment are inclined to think that sociability should be considered as essential for team development.

Finally, we found some empirical evidence to support that depiction of social structures should be useful for the investigation of cooperative patterns and identification of productive social structures. In general, the evidence in this study enables us to understand the social capital embedded in a social network or an organization where strong and weak ties are correlated with social capital. It can be considered as like a hidden bonding that are connecting people to be recognized by a group of individuals. At any cooperation level, we confirm that individuals, who would like to act as a part of the team play, certainly, believes accomplishment of mutual trust is an important ingredient for the success of a social structure

RESEARCH LIMITATIONS

The limitation of this study can be identified as follows. First, we conduct main body of this research by using a university environment. An industrial counterpart, however, should be conducted to confirm our findings, where several measurements may be found different from university settings. In addition, the data collection group was also constrained by its sample size and industrial experience of survey participants. Therefore, an enhanced version of this study should be conducted in an industrial setting based on a medium size software company, in which validity and generalization of initial results can be further examined.

Figure 10. A partial view of a social network of participants



CONCLUSION

Productivity should be considered as a multi-dimensional concept that needs to be carried out from both sociological (DeMarco & Lister, 1999), and economical (Boehm, 1981) perspectives. In this paper, we introduce several definitions regarding to social and value dynamics of a software development organization, and survey the literature to highlight several factors affecting productivity of software development organizations with respect to the value dynamics and several forms of capital. Based on the literature, we identify several factors affecting these constructs and conduct focus groups to choose appropriate indicators from an industrial perspective.

Furthermore, based on these factors we hypothesize three SEM models. In order to validate these models empirically, we create a survey instrument for the data collection process from a university environment. In addition to that effort, we ask participants to rank some of the factors regarding to their importance. Through its ability to illustrate a social structure, a social network analysis is conducted to visualize the existing social structure of participants. This survey is also conducted so as to identify the location of participants' in a social structure. Finally, we attempt to measure the correlation among social productivity, productivity, and social capital of software development.

We found evidence that the identified factors are associated with the defined latent variables. However, all SEM analyses carried out in this paper should be considered as preliminary approaches for establishing a relation between software productivity and social capital. We initially identify that productivity and social capital, and social productivity should be a part of a bigger construct, however, the results should be compared with the data gathered from an industrial setting. We also conclude that these constructs, however, deserves additional examination especially by considering team configurations and effects of personality types of participants. Our next goals are (i) to improve our ability to measure social capital, and (ii) to gather more information by conducting an updated version of our survey on a software development environment.

The process of identifying social factors that contribute to overall well-being of a software company can also help the transformation process of actual resources (e.g., knowledge, team skills, technology) into intellectual assets such as software artifacts. As previously explained, social capital is a form of a tacit or potential knowledge-based resource structured in a network of relations (Lin, 2002). On the other hand, social productivity is particularly concerned with consequences of social behaviors such as results of a team's functioning, and in addition, it should be helpful to investigate the effects of social actions and activities (Barnett, 2004).

Finally, we confirm that the investigation for the transformation process of social capital into software productivity should have some definite advantages especially for understanding the project challenges that IT professionals' encounter. Although, IT professionals who work in software development business regularly assess their skills to strengthen their professional development, they usually focus on advancing their knowledge capital only by learning about the latest technology trends. However, our results strongly suggest that social productivity becomes more important than those individual abilities.

Here, it is noteworthy to mention that improving economic production has an important meaning for software production productivity. Today, this productivity improvement not only depends on personal productivity, effective task planning, task success, but also social characteristics of individuals, and mutual trust in teams and organizations. Thus, in any development process, it is not surprising that social productivity should be observable through collaborative teamwork. Therefore, IT professionals should not only be technically competent but also they need to develop more social and communicational skills. Ultimately, they need to be more efficient team players and more socially productive. In other words, they need to seek out for the best possible configurations to transform their social capital into social productivity. In addition, to avoid productivity losses, we claim that social and structural maturity of software development organization needs to be addressed by understanding the configuration patterns of productive teams. Ultimately, a better team configuration should mitigate the loss of productivity of a software team and organization.

ACKNOWLEDGMENT

This work is supported, in part; by Science Foundation Ireland grant number 03/CE2/I303-1 to Lero, the Irish Software Engineering Research Centre (http://www.lero.ie).

REFERENCES

Abdel-Hamid, T. K. (1996). The slippery path to productivity improvement. *IEEE Software*, *13*, 43–52. doi:10.1109/52.526831

Anderson, D. (2004). *Agile management for software engineering: applying the theory of constraints for business results*. Upper Saddle River, NJ: Prentice Hall.

Baldwin, C., & Clark, K. (2000). *Design rules: The power of modularity*. Cambridge, MA: MIT Press.

Barabasi, A., & Frangos, J. (2003). *Linked: The new science of networks*. Jackson, TN: Perseus Books.

Barnett, L. (2004). Social productivity, law, and the regulation of conflicts of interest in the investment industry. Cardozo Public Law. *Policy and Ethics Journal*, *3*, 793.

Bavelas, A. (1950). Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, *22*(6), 725–730. doi:10.1121/1.1906679

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, *50*(9-10), 860–878. doi:10.1016/j.infsof.2007.09.004

Biffl, S., Boehm, B., & Erdogmus, H. (2006). *Value-based software engineering*. Berlin, Germany: Springer-Verlag. doi:10.1007/3-540-29263-2

Blackburn, J. D., Scudder, G. D., & Van Wassenhove, L. N. (1996). Improving speed and productivity of software development: A global survey of software developers. *IEEE Transactions on Software Engineering*, *22*(11), 875–885. doi:10.1109/32.553636

Blanche, M., Durrheim, K., & Painter, D. (2008). *Research in practice: Applied methods for the social sciences*. Cape Town, South Africa: University of Cape Town Press.

Boehm, B. (1981). *Software engineering economics*. Upper Saddle River, NJ: Prentice Hall.

Boehm, B. (1987). Improving software productivity. *Computer*, *20*(9), 43–57. doi:10.1109/MC.1987.1663694

Boehm, B. (2002). Managing software productivity and reuse. *Computer*, *32*(9), 111–113. doi:10.1109/2.789755

Boehm, B. (2003). Value-based software engineering: reinventing. *SIGSOFT Software Engineering Notes*, *28*(2), 3. doi:10.1145/638750.638775

Boehm, B., & Sullivan, K. J. (2000). Software economics: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland (pp. 319-343).

Bollen, K. A. (1989). *Structural equations with latent variables*. New York, NY: Wiley-Interscience.

Bordieu, P. (1986). The forms of capital. *Handbook of Theory and Research for the Sociology of Education*, 241, 241-258.

Bourdieu, P., & Wacquant, L. (1992). *An invitation to reflexive sociology*. Chicago, IL: University of Chicago Press.

Brynjolfsson, E. (1993). The productivity paradox of information technology. *Communications of the ACM*, *36*(12), 66–77. doi:10.1145/163298.163309

Chemuturi, M. (2009). *Software estimation best practices, tools & techniques: A complete guide for software project estimators*. Ft. Lauderdale, FL: J. Ross.

Chiang, I., & Mookerjee, V. (2004). Improving software team productivity. *Communications of the ACM*, *47*(5), 89–93. doi:10.1145/986213.986217

Churchville, D. (2008). *Agile thinking: Leading successful software projects and teams*. San Diego, CA: Extreme Planner Software.

Cluts, M. (2003). The evolution of artifacts in cooperative work: constructing meaning through activity. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work* (pp. 144-152).

Coleman, J. (1988). Social capital in the creation of human capital. *American Journal of Sociology*, *94*(1), 95–120. doi:10.1086/228943

Coleman, J. (1994). *Foundations of social theory*. Boston, MA: Belknap Press.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268–1287. doi:10.1145/50087.50089

de Barros Sampaio, S. C., Barros, E. A., de Aquino, G. S., Jr., Silva, M. J. C., & de Lemos Meira, S. R. (2010). A review of productivity factors and strategies on software development. In *Proceedings of the Fifth International Conference on Software Engineering Advances* (pp. 196-204).

DeMarco, T., & Lister, T. (1999). *Peopleware: productive projects and teams*. New York, NY: Dorset House.

Dittrich, Y., Floyd, C., & Klischewski, R. (2002). *Social thinking-software practice*. Cambridge, MA: MIT Press.

Fairley, R. (2009). *Managing and leading software projects*. Los Alamitos, CA: Wiley-IEEE Computer Society. doi:10.1002/9780470405697

Fukuyama, F. (1996). *Trust: The social virtues and the creation of prosperity*. New York, NY: Free Press.

Gilb, T., & Finzi, S. (1988). *Principles of software engineering management*. Reading, MA: Addison-Wesley.

Halling, M., Biffl, S., & Grunbacher, P. (2004). The role of valuation in value-based software engineering. In *Proceedings of the 6th International Workshop on Economics-Driven Software Engineering Research* (pp. 7-10).

Hamid, T. K. A., & Madnick, S. E. (1989). Lessons learned from modeling the dynamics of software development. *Communications of the ACM*, *32*(12), 1426–1438. doi:10.1145/76380.76383

Hantos, P., & Gisbert, M. (2000). Identifying software productivity improvement approaches and risks: construction industry case study. *IEEE Software*, *17*(1), 56. doi:10.1109/52.819968

Hayduk, L. (1987). *Structural equation modeling with LISREL: Essentials and advances*. Baltimore, MD: Johns Hopkins University Press.

Hazzan, O., & Dubinsky, Y. (2008). *Agile software engineering*. New York, NY: Springer.

Homans, G. (1958). Social behavioral exchange. *American Journal of Sociology*, *63*(6), 597–606. doi:10.1086/222355

Hoyle, R. (1995). *Structural equation modeling: Concepts, issues, and applications*. Thousand Oaks, CA: Sage.

Hoyle, R., & Smith, G. (1994). Formulating clinical research hypotheses as structural equation models: A conceptual overview. *Journal of Consulting and Clinical Psychology*, *62*(3), 429–440. doi:10.1037/0022-006X.62.3.429

Jackson, M. (2007). The economics of social networks. In *Proceedings of the Ninth World Congress on Advances in economics and econometrics: Theory and Applications* (Vol. 1, pp. 1-56).

Jackson, M., & Petersson, P. (1999). Productivity–an overall measure of competitiveness. In *Proceedings of the Second Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium (pp. 573-581).

Jones, C. (2007). *Estimating software costs: Bringing realism to estimating*. New York, NY: McGraw-Hill.

Jones, C. (2009). *Software engineering best practices: Lessons from successful projects in the top* companies. New York, NY: McGraw-Hill/Osborne Media.

Joreskog, K., & Sorbom, D. (2001). *LISREL8: Users reference guide*. Lincolnwood, IL: Scientific Software International.

Kelly, A. (2008). *Changing software development: Learning to become agile*. New York, NY: John Wiley & Sons.

Kline, R. (2010). *Principles and practice of structural equation modeling*. New York, NY: Guilford Press.

Koh, S., & Maguire, S. (2009). *Information and communication technologies management in turbulent business environments*. Hershey, PA: Information Science Reference. doi:10.4018/978-1-60566-424-8

Kratzer, J., Leenders, R., & VanEngelen, J. (2009). The social network among engineering design teams and their creativity: A case study among teams in two product development programs. *International Journal of Project Management*, *28*(5), 428–436. doi:10.1016/j.ijproman.2009.09.007

Krueger, R., & Casey, M. (2009). *Focus groups: A practical guide for applied research*. Thousand Oaks, CA: Sage.

Lazer, D. (2001). The co-evolution of individual and network. *The Journal of Mathematical Sociology*, *25*(1), 69–108. doi:10.1080/0022250X.2001.9990245

Lim, W. (2002). Effects of reuse on quality, productivity, and economics. *IEEE Software*, *11*(5), 23–30. doi:10.1109/52.311048

Lin, N. (2002). *Social capital: A theory of social structure and action*. Cambridge, UK: Cambridge University Press.

Lovric, M. (2011). *International encyclopedia of statistical science* (1st ed.). New York, NY: Springer. doi:10.1007/978-3-642-04898-2

Marx, K. (1889). *Capital: a critical analysis of capitalist production*. Appleton.

Maxwell, K. (2002). *Applied statistics for software managers*. Upper Saddle River, NJ: Prentice Hall.

Misterek, S., Dooley, K., & Anderson, J. (1992). Productivity as a performance measure. *International Journal of Operations & Production Management*, *12*(1), 29–45. doi:10.1108/EUM0000000001294

Narayan, D., & Cassidy, M. (2001). A dimensional approach to measuring social capital: Development and validation of a social capital inventory. *Current Sociology*, *49*(2), 59–102. doi:10.1177/0011392101049002006

Parnas, D. L., & Clements, P. C. (1986). A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, *12*(2), 251–257.

Pfleeger, S. (1991). Model of software effort and productivity. *Information and Software Technology*, *33*(3), 224–231. doi:10.1016/0950-5849(91)90137-Z

Ping, R. (2004). On assuring valid measures for theoretical models using survey data. *Journal of Business Research*, *57*(2), 125–141. doi:10.1016/S0148-2963(01)00297-1

Portes, A. (1998). Social capital: its origins and applications in modern sociology. *Annual Review of Sociology*, *24*(1), 1–24. doi:10.1146/annurev.soc.24.1.1

Prokopenko, J. (1987). *Productivity management: a practical handbook*. Geneva, Switzerland: International Labour Office.

Scacchi, W. (1995). Understanding software productivity. *Software Engineering and Knowledge Engineering: Trends for the Next Decade*, *4*, 273–316. doi:10.1142/9789812798022_0010

Selby, R. (2007). *Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research*. Los Alamitos, CA: Wiley-IEEE Computer Society.

Shariq, S. (1998). Sense making and artifacts: an exploration into the role of tools in knowledge management. *Journal of Knowledge Management*, *2*(2), 10–19. doi:10.1108/13673279810249341

Sharp, H., Baddoo, N., Beecham, S., Hall, T., & Robinson, H. (2009). Models of motivation in software engineering. *Information and Software Technology*, *51*(1), 219–233. doi:10.1016/j.infsof.2008.05.009

Sink, D., Tuttle, T., & Shin, S. (1989). *Planning and measurement in your organization of the future*. Norcross, GA: Industrial Engineering and Management Press.

Steiner, I. D. (1972). *Group process and productivity*. New York, NY: Academic Press.

Stober, T., & Hansmann, U. (2009). *Agile software development: Best practices for large software development projects*. Berlin, Germany: Springer-Verlag.

Tangen, S. (2002). Understanding the concept of productivity. In *Proceedings of the 7th Asia-Pacific Industrial Engineering and Management Systems Conference*, Taipei, Taiwan (pp. 18-20).

Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and Performance Management*, *54*(1), 34–46. doi:10.1108/17410400510571437

Tsui, F. (2004). *Managing software projects*. Burlington, MA: Jones & Bartlett Learning.

Wasserman, S., & Galaskiewicz, J. (1994). *Advances in social network analysis: Research in the social and behavioral sciences*. Thousand Oaks, CA: Sage.

Welfe, W. (2009). *Knowledge-based economies: models and methods*. New York, NY: Peter Lang.

Yilmaz, M., & O'Connor, R. (2011). An empirical investigation into social productivity of a software process: An approach by using the structural equation modeling. *Systems*. *Software and Service Process Improvement*, *172*, 155–166. doi:10.1007/978-3-642-22206-1_14