An Empirical Investigation into Social Productivity of a Software Process: An approach by using the structural equation modeling

Murat Yilmaz[1] and Rory V. O'Connor[2,3]

[1] Lero Graduate School in Software Engineering, Dublin City University, Ireland
murat.yilmaz@computing.dcu.ie
[2] Dublin City University, Ireland
[3] Lero, the Irish Software Engineering Research Centre
roconnor@computing.dcu.ie

Abstract. The actual and expected benefits of fostering the alignment of people factors and cooperation among software development teams enables software development organization to improve software development productivity. Furthermore, software development productivity presents a significant challenge for both understanding and quantifying the performance characteristics of software development organizations. This paper introduces an approach to model software development productivity by using structural equation modeling (SEM), a technique that can be used for testing and estimating relationships using empirical data. We also present preliminary results from an exploratory study about the enabling social factors that affect software development productivity. Our quantitative analysis involves grouping productivity and social productivity factors for studying and identifying their implicit relationship. To this end, we issue questionnaires to test our hypothesis and to gather sample data. The paper concludes by showing initial results, limitations, and directions for future research.

1 Introduction

In contrast to other aspects of software process improvement (SPI), software productivity improvement is a multi dimensional concept with a means of achieving and sustaining a competitive advantage. As software development is considered to be a human endeavor (i.e. effort and intellectually intensive team work) [1], the human and social aspects of software engineering has turned into an important topic to investigate for both scholars and practitioners. It is therefore not surprising 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 individual and a team as a whole. Although extensive research has been carried out about many of the social aspects [2], no study exists which adequately uncovers the relationship between the productivity factors and the

social aspects affecting productivity of software development projects. We therefore conduct a study of this relationship and in doing so, we identify an indicator for defining the social aspects influencing productivity, we term this indicators as *social productivity of software development*. Social productivity is a dimension of productivity which addresses improvement issues about social interactions corresponding to the basis of the social structure of a software team.

1.1 Objective of Research

Our preliminary study aims to empirically analyze the dynamic interactions between the factors of social productivity and productivity based upon software productivity literature and refined by our focus group studies. We apply a structural equation model (SEM) approach for evaluation of the central hypothesis (i.e. a positive correlation between influential factors of social productivity and software productivity) introduced in this paper to be tested and to provide an initial empirical support for our proposed model. One dimension of the model consists of several important productivity factors found in the literature, where other dimensions define several indicators of the social aspects of software productivity. Most importantly however, this paper is among the first attempts to use a technique like SEM to examine the impact of social aspects of software development productivity.

The remainder of this paper is organized as follows: In the next section, we introduce several definitions for the social dynamics of software development settings. The following section describes the analysis models and methods in more detail and presents some some preliminary results that prove the feasibility of our proposed model, and to verify our empirical approach. Finally, the last section concludes the paper with a brief summary of contributions and the directions for future research.

2 Social Dynamics

Social dynamics is an multi disciplinary field of science that concerns the process of analyzing socialites or social systems expressed by actors and their interactions based on rules or norms. These definitions provided by the field of social dynamics help us to highlight the important points of the Social Aspects of Software Engineering (SASE) [2]. Ultimately, SASE will help us to understand the social dynamics of a software organization in order to promote cooperation within software teams and organizations, and to respond to the dynamic trends of present and future of software development.

2.1 Productivity

Software production is the economic process of conversion of inputs to outputs based on resource consumption and allocation. Thus, one of the concerns of software process improvement is investigating methods to improve and measure

the software productivity. In general, productivity is a value to measure the efficiency of this production process.

A common definition of software productivity from the literatue is the 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 [3]. However, it is hard to find a suitable way for measuring productivity [4] because, it may be considered differently for stakeholders from their distinctive perspectives. For example; from the viewpoint of developers, a productivity measure would be the amount of code produced for the software system, on the other hand from the user's perspectives; it could be the the degree of functionality achieved for the software system. An increase in the productivity is achieved when activities and resources in the software development process are use to add more value to the software product.

During several development activities multiple outputs are produced concurrently, therefore Scacchi [5] suggest that a multi dimensional analysis of productivity is important in software development settings. Productivity can therefore considered to be a multi dimensional problem significantly affected by many factors including the quality of workforce, management capabilities and environmental conditions of a software organization. However, the social factors of software productivity can't easily be identified, e.g. cost of communication and social expenses [6].

Software engineering researchers suggest methods to improve software productivity by balancing the demands of people (regarding to their activities), processes (with respect to their tasks) and technology (by its advances in computing power) [7, 5, 8].

One common approach for improving software productivity relies on the theory of group productivity introduced by psychologist Ivan Steiner [9]. It states that consequences of defective processes are important for explaining actual productivity. Abdel-Hamit [10] defines the notion of potential productivity where maximum productivity is only achieved if an individual or a team uses their maximum potential. He added two factors that are important for representing the shortfalls for software quality and productivity problems; (i) task 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.

Over the past few decades, software productivity has been investigated by using several indicators affecting the productivity. One such approach is conducted by Pfleeger [11] who uses 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 [12].

Finally, productivity improvements can be achieved by having a skillful team, improving the path of development by reducing rework, and by creating reusable and more manageable software artifacts [13].

2.2   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. Social Capital can be defined as the capital which is attracted and held by social connections and networking so as to make a gain or profit. Lin [14] defines social capital as an "*investment of social relations with expected returns in [the] market place*". Bourdieu [15] defines the term social capital as a mass of present and future resources that are linked as a network of relationships. His definition designates that social capital is based on two components; (i) social relationships which affords possibilities to help them obtaining accessibility to the resources by their relationships, and (ii) resource quality.

Social captial can be seen as an-other resource to be captured by individuals [16]. According to Portes [17] social capital is inherent in the fabric of actors and relationships. In order to own social capital, one should have linked with others. Therefore, social capital should be measured somehow with respect to the quantity and quality of social connections that one might have. Coleman [18] argues that all kinds of social structures and relations enable some form of social capital. As a matter of fact, individuals intentionally connect with one and other to form social networks and expect benefits from these actions.

The level of social capital attainable by participants of a software development organization will ensure the enthusiasm of teams and individuals to cooperate in a voluntary manner. Social capital should help to improve the social coordination and stability. Therefore, it will enable us to have an efficient information exchange network [16] .

2.3   Social Productivity

In the socio-economic landscape of software organizations, increasing the efficiency and productivity of individuals and organization by improving their social capital depends on the subset of various facts or several circumstances (e.g. quality of social interactions). The act of understanding the impact of social relations in process, tasks and activities of development can be considered as an important aspect of productivity.

We define, **social productivity** *as the production rate of software development increases if we give due consideration to maximizing the social relations.*

Therefore, we claim it is important to understand the concept of social productivity as a measure (level) for collaborative outcome by social interactions through a software company. Accordingly, this quantification can be use to improve the positioning of the teams and individuals in software organizations. It also can help the transformation process of actual resources (e.g. human knowledge, team skills, time, technology) into assets (i.e. software artifacts).

The notion of *social productivity of software development* aims to highlight the social outputs of organized groups and the importance of interactions and

behaviors. It may have some beneficial usage for measuring values like cooperativeness or some other harmful formations (e.g. conflict of interests) for not only for the software teams but also for the entire software development organization.

3   Models and methods

This section describes models and methods that are used in our investigation of social productivity factors. First, we introduce the structural equation modeling which our productivity model is based on. Next, we highlight the benefits of a focus group and explain our effort to identify the factors of productivity in an industrial setting. Further, we depict our model for software productivity and the framework that we used to conduct the research.
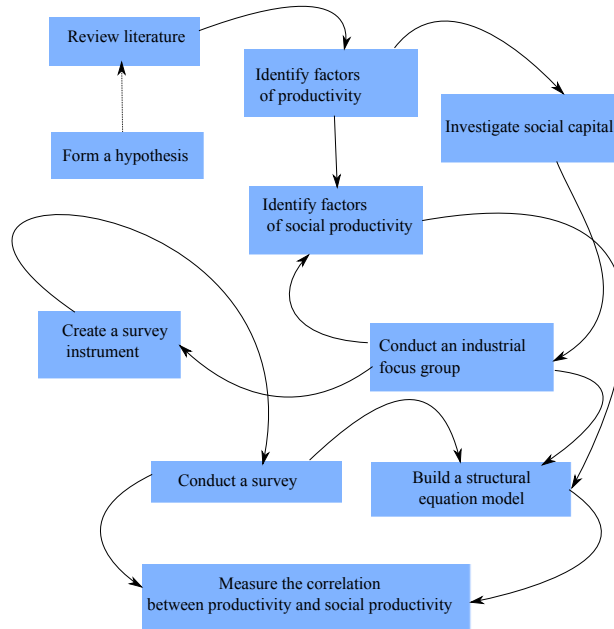
3.1   The Systematic Approach

Here, we develop a systematic approach to address the relationship between productivity and social productivity (see figure 1). First, based on a structural equation model, we formed a hypothesis which states that social productivity is highly correlated with productivity. Second, we reviewed the software productivity literature to investigate the factors affecting productivity. After identification of several productivity factors, we proposed a set of factors affecting social productivity. To evaluate this proposal, third, we conducted a focus group research and consulted a software company for their opinion about these identified factors, and consequently utilize this information to change some our initial settings. Fourth, we create a survey instrument for testing and validating the causal relationships we proposed among several factors (i.e. observable and latent variables) and so as to refine the structural equation model. Finally, to identify and to examine causal relationships among several factors that are affecting the quality of software development, we conducted a survey by using graduate and post graduate university students.

3.2   Structural Equation Modeling

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) [19]. Based on patterns of statistical expectation, it is a confirmatory multivariate (multi equation) analysis technique for estimating the structural or casual relationship among the variables that are observed and latent, and specifying relations among these latent variables [20].

SEM models use a collection of simultaneous equations, which are based on a combination of observed and latent variables (hypothetical constructs or factors), which are introduced and frequently used by sociology [21] and psychology [22] research and econometric methods [23].

Fig. 1. The systematic approach for software productivity research

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 [24].

3.3 The Measurement Model

We chose productivity and social productivity as latent variables (i.e. one type of factor) for our structure equation model. Although some approaches address productivity as a construct, no previous study has been found on social productivity factors and its relationships with productivity of software development.

Our model is based on factors affecting productivity and social productivity. By using the productivity literature in general, and software development productivity in particular we chose five factors that have been mostly referenced by researchers. The initial factors we found important were; (i) Motivation, (ii) Process, (iii) Reuse, (iv) Complexity, and (v) Team Size. Next, we aimed to use four observed variables including; (i) Leadership, (ii) Trust, (iii) Communication, (iv) Team Cohesion for the measurement of social productivity.

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. However, we argue that a focus group is an efficient way to reach that information [25]. The group setting may be ideal for people to build new ideas on the

top of other's opinions and further discussing their experiences [26]. 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 the CEO of the company (total ten participants). As suggested by Krueger [25], the session was facilitated by one of the authors who commenced an introduction to encourage participants and initiate the discussion setting. 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: (1) What is your definition of productivity in software teams?, (2) What is your opinion of the factors that are affecting the productivity?, (3) What do you think of the most important factors among these ones for productivity?, (4) How would you describe the social factors of productivity?, (5) What is your opinion of the social factors that are affecting the productivity?

The goal of the focus group study was to identify the opinions from industry about the most important factors that are affecting for both productivity and social productivity. 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, 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, i.e. skills and reuse. In short, focus group activity provides us an opportunity to discuss our ideas about productivity factors in an industrial setting. We refined our list of factors by using the information provided in this session.

In light of these results, we designed a survey instrument to measure the impact of the factors on both productivity and social productivity. We used 5-point Likert scale (i.e. a psychometric scale frequently used in social research) for every factor and furthermore we add two question where they were asked to rank their opinions in descending order of importance for productivity and social productivity factors.

3.4 Software Productivity as Linear Structural Relation

A generally accepted measurement model of productivity is lacking [27], hence, we suggest that productivity and social productivity can be presented as latent variables showing themselves through a set of factors. In addition, we argue that these variables also influence each other. Based on the several important factors affecting both variables, we draw a model of social productivity by using SEM and aim to specify their interrelationships (see figure 2) for a conceptual

representation of the hypothesized model). The observed variables are shown 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.
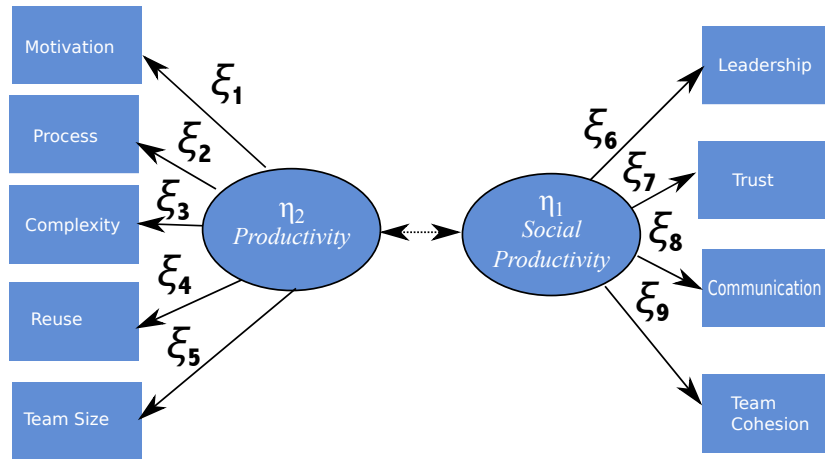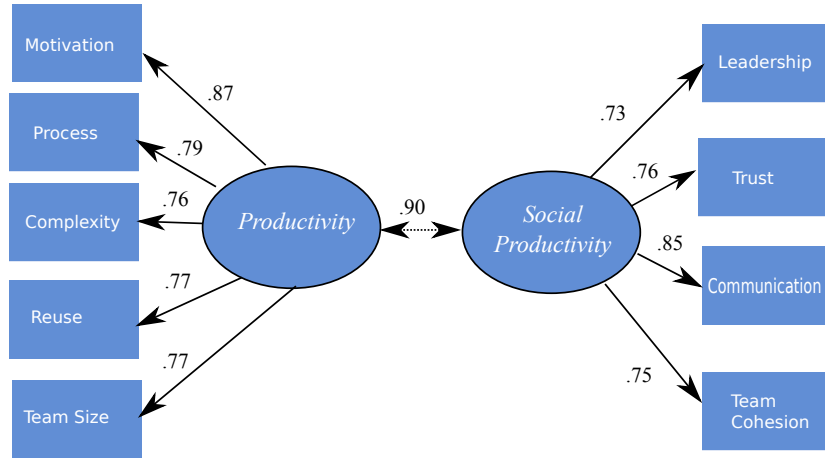

Fig. 2. Conceptual Model for Social Productivity of Software Development

For modeling social productivity and productivity factors, we use LISREL [28] (i.e. a software package frequently used for structural equation modeling) and proposed them as latent variables based on four and five observed variables respectively (nine indicators in total). The data was collected by surveys obtained from graduate and undergraduate university students. The analyses was conducted with 200 participants. About 67 percent of the participants were post graduates. In this work, we used a two-step approach [24], first we explored the measurement model which specifies the relationships between indicators and latent variables used. Secondly, using the results of the measurement model we test the structural equation model for an acceptable goof fit. We suggest that all factors should be interacting with each other. The latent variables namely, social productivity and productivity, are bivariate correlated. The hypothesized model is presented in figure 2 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.

The analyses were conducted with 200 participants. Two stages of data analyses were conducted to test the conceptual model depicted above. First, for testing the measurement model and second, for structural model. We developed a maximum-likelihood method for fitting the mathematical model to collected data and for the estimation of model parameters. A null hypothesis (i.e. independence model) which assumes variables among the relationships are uncorrelated (i.e. worst case scenario) is totally rejectable where $\chi^2(95, N = 200) =$

Fig. 3. Structural Equation Model for Social Productivity of Software Development

$2123.32, p < .001$. Consequently, the measurement model (see figure 2) was found to differ a good fit for the data $\chi^2(82, N = 200) = 3983.71, p < .001$, where $RMSEA = .081, GFI = .99, AGFI = .91, CFI = .99, NFI = .98$), where all of the structural correlations between latent variables were statistically significant ($p < .05$) and ranged between *Motivation* (Structural Coefficient= .87, $p < .05$) and *Leadership* (Structural Coefficient= .73, $p < .05$). Motivation has the strongest relationship with productivity among all other factors, while *Communication* has the strongest connection between social productivity. It has been suggested that a chi-square difference test indicated significant improvement in fit between the independence model and the hypothesized model, $\Delta\chi^2(26, N = 200) = 3983.71, p < .001$).

In the next step of the analysis, we refine our structural model to include social capital as an additional latent variable (see figure 4), and therefore we add new indicators affecting the social capital including; (i) communication transparency, (ii) social relations, (iii) frequency of meetings. The refined model was tested a good fit for the data, $\Delta\chi^2(26, N = 200) = 3983.71, p < .001), RMSEA = .15, CFI = .84, AGFI = .75, CFI = .95, NFI = .94$). Results of a chi-square difference test indicated that measurement model and hypothesized model are identical, $\Delta\chi^2(3, N = 200) = 3.71, p < .05$. According to the path diagram, it seems collective outcomes, information awareness and mutual trust are significant predictors of social productivity improvement. Moreover, it is obvious that social capital is one of the major ingredients of social productivity (Structural Coefficient = .89, p < .05), which is also very highly correlated with the productivity (Structural Coefficient = .86, p < .05).
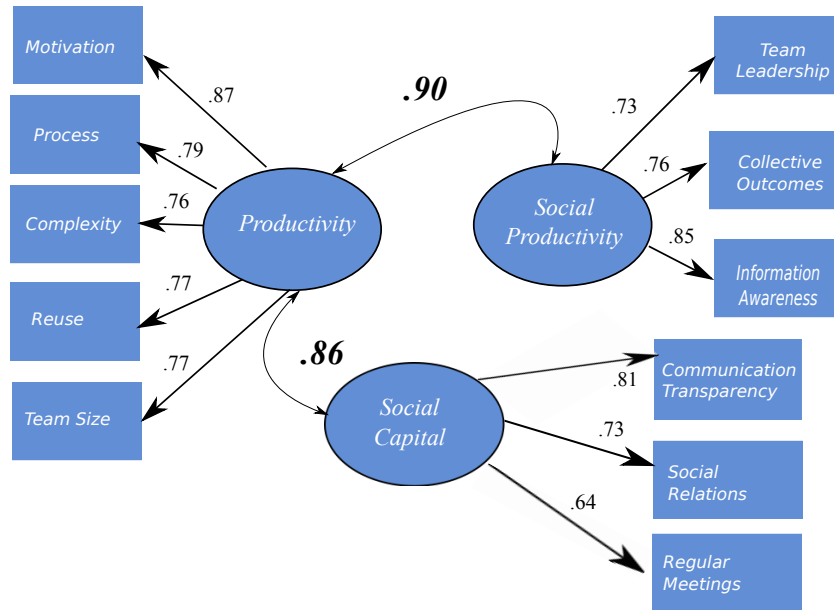
Fig. 4. Structural Model for Social Capital of Software Development

4 Conclusions and Future Work

In this paper, we propose an empirically validated model to measure the correlation between social productivity and productivity of software development. The evaluation of indicators are discussed with a focus group study by collaborating with the management team of a medium-sized software company. In addition, a survey instrument is created and tested with survey data collected from 200 graduate and post graduate students (most of which have an industrial experience). Consequently, we constitute an initial model with 9 structural path relations. The goal of a structural equation model in this context supports the following outcomes. First, the observational results will provide insight into how different factors are affecting both productivity and social productivity. Second, by refining the first form, an improved model with 11 path relations is designed by using the notion of social capital.

We have only found a single study which investigates information system productivity based on structural equation modeling that may be related to our research. Based on participants from a Hong Kong information technology organizations, Foulds *et al.* [29] used structural equation modeling for developing and testing a framework for the productivity of large scale information system development. Their results show that better product descriptions and a dynamic approach to project management have a positive impact on system development productivity.

This study confirms that our approach should be useful for software productivity research for several reasons. First, we propose a linkage of structural equation modeling and SPI. In general, we suggest that, this approach can be useful for correlating latent (qualitative) variables and observable variables where empirical data can be collected. Consequently, the factors of interest can be revealed which aids managerial decision support. Second, we introduce the concept of social productivity and examined causal factors affecting productivity (leadership, team cohesion, collective outcome, trust) and identify their importance with respect to the opinion of our survey participants. Third, we introduce three variables to measure social capital of software development organizations (social relations, frequency of team meetings, interaction efficiency). Furthermore, we calculate several correlation values for factors investigated in both of our models.

SEM is a modeling method frequently used to solve several problems encountered in social sciences. Our first structural model indicates that there is not only a significant amount of correlation between productivity and social productivity but also a correlation occurs among their interacting factors. Therefore in the refined model of productivity, we introduced social capital as a new latent variable and formalized our second model based on these facts.

By modeling various aspects of productivity using a structural model, a researcher can obtain clear insights into the factors that are affecting productivity. In light of this, our research makes a valuable contribution to the practice of software productivity improvement. Our next goal is to conduct the survey to evaluate our model on several software companies for comparison with our initial results.

Acknowledgments

References

1. Fairley, R.: Managing and Leading Software Projects. Wiley-IEEE Computer Society Pr (2009)
2. Dittrich, Y., Floyd, C., Klischewski, R.: Social thinking-software practice. The MIT Press (2002)
3. Chemuturi, M.: Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimators. J. Ross Publishing (2009)
4. Tangen, S.: Understanding the concept of productivity. In: Proceedings of the 7th Asia-Pacific Industrial Engineering and Management Systems Conference, Taipei. (2002) 18–20
5. Scacchi, W.: Understanding software productivity. Software Engineering and Knowledge Engineering: Trends for the Next Decade (1995) 273–316
6. Abdel-Hamid, T.K.: The slippery path to productivity improvement. **13** (1996) 43–52

7. Gilb, T., Finzi, S.: Principles of software engineering management. Addison-Wesley Wokingham, UK (1988)
8. Hantos, P., Gisbert, M.: Identifying software productivity improvement approaches and risks: construction industry case study. IEEE Software **17** (2000) 56
9. Steiner, I.D.: Group process and productivity. Academic Press New York (1972)
10. Hamid, T.K.A., Madnick, S.E.: Lessons learned from modeling the dynamics of software development. Commun. ACM **32** (1989) 1426–1438
11. Pfleeger, S.: Model of software effort and productivity. Information and Software Technology **33** (1991) 224–231
12. Maxwell, K.: Applied statistics for software managers. Prentice Hall PTR (2002)
13. Boehm, B.: Improving software productivity. Computer **20** (1987) 43 –57
14. Lin, N.: Social capital: A theory of social structure and action. Cambridge Univ Pr (2002)
15. Bordieu, P.: The forms of capital. Volume 241. (1986)
16. Coleman, J., Coleman, J.: Foundations of social theory. Belknap Press (1994)
17. Portes, A.: Social capital: its origins and applications in modern sociology. Annual review of sociology **24** (1998) 1–24
18. Coleman, J.: Social capital in the creation of human capital. ajs **94** (2007) S95
19. Hayduk, L.: Structural equation modeling with LISREL: Essentials and advances. Johns Hopkins Univ Pr (1987)
20. Lovric, M.: International Encyclopedia of Statistical Science. 1st edition. edn. Springer (2011)
21. Bielby, W., Hauser, R.: Structural equation models. Annual Review of Sociology **3** (1977) 137–161
22. Bentler, P.: Multivariate analysis with latent variables: Causal modeling. Annual review of psychology **31** (1980) 419–456
23. Aigner, D., Hsiao, C., Kapteyn, A., Wansbeek, T.: Latent variable models in econometrics. Handbook of econometrics **2** (1984) 1321–1393
24. Kline, R.: Principles and practice of structural equation modeling. The Guilford Press (2010)
25. Krueger, R., Casey, M.: Focus groups: A practical guide for applied research. Sage (2009)
26. Templeton, J.: The focus group: a strategic guide to organizing, conducting and analyzing the focus group interview. Irwin (1994)
27. Jones, C.: Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies. McGraw-Hill Osborne Media (2009)
28. Joreskog, K., Sorbom, D.: LISREL 8: users reference guide. Lincolnwood, IL: Scientific Software International. Inc (2001)
29. Foulds, L., Quaddus, M., West, M.: Structural equation modelling of large-scale information system application development productivity: the Hong Kong experience. In: Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on, IEEE (2007) 724–731