Who Are the Rational Actors in Software Development Projects?

Cornelia Gaebert[1] and Jörg Friedrich[2]
[1]Research Group on Strategic Information Management, European Research Center for Information Systems
c/o University of Muenster, Leonardo Campus 11, D-48149, Muenster, Germany
[2]Kultur- und Sozialwissenschaften, Lehrgebiet Philosophie III, FernUniversität in Hagen, D-58084, Hagen, Germany

Keywords: Group Agents, Collective Actions, Rationality, Software Development Project, Software Development Outsourcing, Software Project Management, Rational Actors.

Abstract: In the field of software development outsourcing and software project management, researchers use concepts from economic theory to describe organizations, groups, teams, and involved people as rational actors. However, they fail to justify these approaches based on an appropriate understanding of the involved social actors' status. The question, who really can be described as a rational actor, the organizations, teams, or the individual people, and which kind of rationality they provide, remains open. We have therefore analyzed the social structure of software development projects (SDP) as described in research literature. Based on novel concepts from the field of social philosophy, we have developed a social ontology (SO) of the actors in the context of commercial SDPs. We identified different actors at several levels, with different kinds of rationality. Organizations, departments, and groups may act as rational actors if following well-defined regulations and methods. Actor classification according to the rationality of the actors' decision-making will help understand and predict their behavior and thus provide a solid base for the application of economic concepts to software development outsourcing and software project management research.

1 INTRODUCTION

During the last few decades, a lot of research regarding software development outsourcing and software project management has been using economic theory approaches in order to understand processes and actors' behavior (cf. Benaroch et al., 2003; Aubert et al., 2003; Beulen and Ribbers, 2003; Lichtenstein, 2004; Gaebert, 2014a). The prerequisite of this approach consists in depicting the actors as rational agents. However, it remains unclear who really is the actor: is it the individual person, the group or the team, or the organization. It also remains unanswered in which sense we can ascribe rationality to these actors. There is a paucity of research about why we are capable of describing individuals, groups, or organizations as rational actors, and on the preconditions of this description.

In this paper, we will analyze the social structures existing in a software development project (SDP), and in this way develop a social ontology (SO) of the part of the world where the SDP is located. We will use the notion of ontology as it is used in social philosophy. As a philosophical discipline, it deals with what the social world is composed of, and with the basic matter and constitution of social reality (Pratten, 2014). Do institutions, collectives the entities constitute the world? Do they really exist in a way, that we cannot reduce them completely to the behavior of the individual people? The social ontology as a philosophical discipline tries to answer such questions.

We use the notion ontology not only for the philosophical discipline reasoning about existence, but also for the result (Lawson 2014). The system of entities for which we are justified to say they exist, is the ontology of (a part of) the social world. In this sense, we will develop in this paper an ontology of the actors in software development projects by analyzing the rationality of these actors.

The aim of this paper is to contribute to clarifying the social structure of SDPs in companies and other organizations, who aim at developing software systems supporting business process. Hence, it introduces an ontological basis for research in software development outsourcing and software project management. We will answer the question, under which conditions we are justified to describe an organization or a group in a SDP as a rational actor. From this, it is possible to answer the question if we

Gaebert C. and Friedrich J..
Who Are the Rational Actors in Software Development Projects?.
DOI: 10.5220/0005516101310138
In *Proceedings of the 10th International Conference on Software Engineering and Applications* (ICSOFT-EA-2015), pages 131-138
ISBN: 978-989-758-114-4

can apply economic theories based on the concept of rational actors to problems in SDPs.

We will develop our argument in four steps. In section 2, we will give a brief literature review of the social structure of SDPs. Then, we will introduce novel relevant approaches of social philosophy such as intentional stance, collective rationality, and joint action in section 3. In the fourth section, we will bring both together and describe SDP actors in terms of social philosophy. In the last section, we will summarize the findings, and we will shortly discuss open questions on SO philosophy. We claim that researchers can better understand processes like decision-making, software production, and stakeholder involvement by using a clear social actor ontology.

2 ACTORS IN SOFTWARE DEVELOPMENT PROJECTS

If a company or an administrative authority meets a new challenge in its business, it often needs a new software system to support the changing or newly implemented business processes. In many cases, it is helpful to sign a contract with a software company aiming to develop the needed system (Bakker et al., 2011, Marschollek and Beck, 2012). Due to the quota of failing SDPs (El Emam and Koru, 2008, Al-Ahmad et al., 2009, Standish Group, 2010), many researchers have been investing a high amount of work over decades to find out the causes of project problems and to develop problem-solving strategies (cf. Dwivedi et al., 2013). These researchers describe and analyze a couple of different actors, their behavior, their interests, and decisions: the contractual partners, the stakeholders, the groups and teams within the SDP, and last but not least the people who perform the tasks. It is obvious that the ontological status and the kind of rationality of these actors must be different.

In this section, we will analyze actor descriptions in the research literature on software engineering issues at different work levels.

2.1 Decisions and Structures inside the Software Development Organization

Yilmaz et al. (2010) analyze the structure of a software development process as being divided into activities and tasks assigned to organizations, teams, or individuals. Their aim is to use economic approaches like the prisoner's dilemma from game theory in order to understand decision-making in SDPs. In an overview of relevant research literature, it is shown that the prisoner's dilemma can be used to describe different interactions in SDPs, from customer-supplier interactions in offshore SDPs and interactions between the test team and the development team to the cooperation of programmers in extreme programming. If we take this point of view, we can consider all actors as rational agents, making their decisions by the use of well-defined strategies aiming at their individual and independent goal, i. e. to maximize their outcome. The authors describe a software company as a social ecosystem consisting of groups of individuals connected by information-based interactions. In such ecosystems, each participant has its individual strategies for dividing its activities into tasks. Following this line of reasoning, we can interpret the organization as a hierarchical network of organizational units which perform tasks or activities to reach specific goals. On each level of the network's hierarchy, the actors make rational decisions to select a task or start an activity in order to reach a defined goal.

Zannier's and Maurer's (2007) empirical research shows a similar representation of the description of decision-making. The authors study the agile and non-agile processes of decision-making in software development companies. The analyzed companies have implemented well-defined rules for evaluating alternatives and for making choices. The paper shows that both in agile and in non-agile working companies decision-making is due to well-established process definitions. Again, we can see that the individual programmer does not necessarily make decisions in the strong sense, when the individual is just following a predefined set of rules. There are, as the study shows, individual decisions based on individual experiences. In addition, there are decisions strongly determined by company regulations. Therefore, these are decisions using the organization's rationality.

2.2 Collective Structures of Software Engineering Processes

An SDP is embedded in a software engineering environment which exists longer than the single project and which involves more people than just the project participants. Consequently, the SDP's social structure is connected and included in many ways in a long-living and extended social structure. Tamburri et al. (2012, 2013) investigated the social structure of software engineering processes. They found four types of collective structures: communities, networks, teams, and groups. Regarding further structure attributes, they discuss thirteen relevant types of so-called organizational social structures (OSS) (Tamburri et al., 2012).

Often, communities and networks are formed without regard to a tangible SDP. Communities are implemented for sharing, maybe of an interest, of knowledge, or experiences. People building a community wish to share knowledge or experiences, because they have a common interest. This sharing may help them do their work inside the SDP, but it does not influence the decision-making directly inside the SDP.

The purpose of a network is reaching other people in a defined way, mostly for information exchange. Networks are also independent from SDPs. They may build the communication infrastructure for the SDP participants. Consequently, they can make faster decisions in SDPs, but at first glance, like communities, they do not influence these decisions directly.

Inside SDPs, we will find teams and groups. Groups are stable parts of an organization, for instance departments or long-term teams which are well defined and structured by the organization's design principles. On the other side, teams are constituted in order to solve a single task, or a limited number of tasks. The project team is a team in the sense of the OSS defined by Tamburri et al. (2012, 2013), but there are also teams inside the SDP, for instance the team designing the system or the test team.

If we use the four concepts for the description of an OSS as suggested by Tamburri et al., we are able to describe the whole complexity of social dependencies between people working inside a SDP. We will use the terminology of these researchers in our following analysis of the SO of SDPs. This will especially help to understand decision-making and, therefore, the rationality of socially constituted actors. We will pick up on this issue again in section four.

2.3 Structures of the Cooperation between Customer and Supplier

Whereas the studies mentioned above investigate the processes within a software developing company, Marschollek and Beck (2012) focus on problems arising during customer-supplier cooperation. The customer needs the software system, and the supplier performs the SDP. The authors analyze how so-called cultural differences between the parties lead to risks and challenges in processing the SDP. As their empirical investigation shows, public authorities, and private enterprises implement extremely different strategies for processing tasks and solving problems. The organization's culture (Kotter, 2008) determines these strategies. Culture defines how participants make their choices for actions. Therefore, from an external point of view, the other organization appears as the unknown, which is acting in a strange way. Problems arise when customer people and supplier people work together in teams for a certain period, and when they have to decide together in these teams.

2.4 Summary: A Three-Level Hierarchy of Actors

Summing up the results of the cited literature, we find a hierarchy of different actors in the SDP. Each of them seems to have a certain kind of rationality for decision-making regarding the achievement of specific goals (Table 1).

Table 1: Actor levels in software development projects.

Level	Actors	Examples
Contract level	Organizations	Supplier, customer
Project management level	Groups	Procurement department, test department
	Teams	Project team, design team
Working level	Individuals	Designer, user, architect, developer

At the highest level, we find organizations. These organizations make decisions in order to sign contracts and to process the SDP's business issues. This decision-making process depends on rules defined within the organizations and also on the culture of each organization.

At the hierarchy's intermediate level, we find stable groups and short-term teams interacting for a certain period inside the SDP. SDP planning and management take place at this level. Therefore, this is the project management level.

The stable groups consist mostly of individuals of only one organization. These individuals also work together outside the SDP, and often for a long time. These groups also follow defined regulations or cultural rules when it comes to organizing their work and making their decisions. Nevertheless, there are also teams consisting of individuals from several organizations, i.e. teams that only exist for a shorter period. They also have certain goals, for instance the specification of requirements. In such teams, we will seldom find an implementation of cultural rules, but there possibly are defined regulations for the cooperation of the team members, determined in the contract or by standardized process models.

Finally, at the lowest level, we find the individuals who do their job. This is the working level. The individuals make rational decisions, sometimes comparatively free, and sometimes strongly ruled by process definitions and regulations. They have beliefs rooted in experience and individual knowledge and use their means to reach desired ends.

These individuals are connected with others inside and outside the SDP by networks. Since they need information from other people when it comes to making decisions, the implementation of such networks will influence the process and the results of decision-making.

Furthermore, the individuals are part of several communities. These communities may influence the individuals' preferences and values. Therefore, also community membership can be of significance for decision-making inside the SDP.

Figure 1 shows our picture of the social structure of an SDP's environment. The blue circles symbolize the individuals, and the lines connecting the individuals symbolize the networks the individuals are involved in.

To understand the social complexity of an SDP, we must analyze how decision-making works at the different levels of the described social structures. In this context, it may be helpful to use the concepts of philosophical SO, because these concepts provide a systematic description of the rationality of social actors. We will introduce these concepts in the next section.

3 SOCIAL ACTORS AND THEIR DECISION-MAKING

In this section, we will briefly provide the basic concepts of SO needed for ascribing an independent status of existence to the social entities described in the previous section. Starting from there, we will derive rationality criteria for these entities.

3.1 Intentional Stance

In descriptions like those depicted in the previous section, we obviously take up the intentional stance described by Dennett (1981). This means that we attribute beliefs, desires and actions to an object, even if it is not an individual person. Strictly speaking, we do not say that these objects really do have beliefs and desires. As Dennett stated, we make propositions about the question concerning which beliefs the object ought to have, and which desires it ought to have, and whether it has some kind of rationality.

Tollefsen (2002) shows that we can take up the intentional stance to describe and interpret organizations. As this author states, if we ascribe responsibilities to organizations like companies or states, we must also accept that these objects are intentional systems as defined by Dennett.

Tollefsen (2004) suggests to closely connect intentionality with rationality. This means that the intentional system has the ability to assess its beliefs with regard to consistency and truth, and that it can derive the right action from these beliefs in order to reach the desired goals by making stable decisions.

Therefore, we can describe an intentional object acting in such a way as a rational actor.

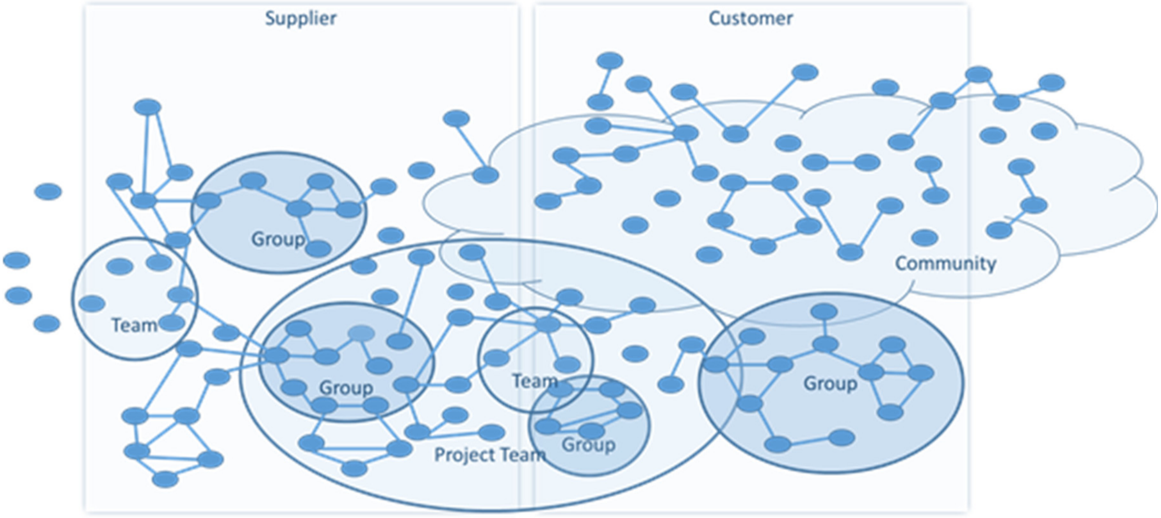

Figure 1: Collective actors in SDPs.

3.2 Organizations as Rational Actors

As Tollefsen shows, in this sense organizations are able - as organizations - to testify to propositions because they can produce stable testimonials and because other organizations and individual persons accepted them as testifiers (Tollefsen, 2011). As Tollefsen shows, organizations have the ability to generate their own rationality through stable methods and regulations for information processing regarding goal definition, belief generation, and decision-making.

If we consider the organization as a rational actor, we suggest that there is a necessary precondition that these methods and regulations be implemented as institutions, according to the signification introduced by Searle (see for instance Searle, 2006). As suggested in section 2.3, we can call these accepted methods and regulations the culture of the organization.

3.3 Conditions of Joint Actions

Nevertheless, not every defined group of people and not even every organization possess such an accepted set of regulations and methods. Thus, there are organizations that are not in a state of rationality (see Gallotti, 2011). Therefore, we cannot attribute them as rational actors (cf. Tollefsen, 2011). Nonetheless, as shown by Pettit and Schweikard (2006), there are other kinds of actions in groups without group intentionality in a strong sense. These authors argue that in the case of joint actions each individual person belonging to the group must have a set of beliefs and intentions.

We must ask how this complex set of beliefs and intentions is possible. Pettit and Schweikard (2006) suggest that we are capable of having the required beliefs because we are social beings. In addition, considering our natural and social history, we have developed an ability to detect such possibilities for joint action. We think this is not sufficient. After having discussed some examples from everyday SDP experience, we will discuss this issue shortly in the last section of the paper.

4 COLLECTIVE RATIONALITY IN SOFTWARE DEVELOPMENT PROJECTS

We are now coming back to the organizations, groups, teams, communities, networks, and individual persons acting within the SDP as announced in section 2 of this paper. We will analyze below how to describe them as rational actors, in particular by analyzing the rationality of their decision-making.

4.1 Suppliers and Customers as Rational Actors

First, it seems obvious that the customer and the supplier, as with companies or public authorities, are organizations in the sense defined by Tollefsen (2002). They do not only act by delivering documents like specifications, by signing contracts, producing and delivering software, and finally by accepting the product and paying the amount of money agreed. All these actions can be interpreted as a testifying of propositions. The customer, as an organization, demands fulfillment of the requirements. The supplier, as an organization, declares that he is able to produce the needed system. In addition, it is the supplier who demands fulfillment of the customer's obligations, and who declares the readiness of delivery. Finally yet importantly, it is the customer who accepts the delivered software system.

Furthermore, we find clear regulations on decision-making in nearly all organizations. Therefore, we can attribute rationality to them. We find mechanisms to generate collective knowledge and to testify propositions regarding the organization itself or its business partners.

Nevertheless, we must be careful. Personal relationships or individual power often bypasses and avoids well-defined regulations in organizations (cf. Narotzky, 2007, Edum-Fotwe and Price, 2009). In this case, it gets harder to attribute rationality to the organization and to describe it as a rational actor. In order to be a rational actor, a true believer, it is not only necessary to produce testimonials, but also to produce them in a stable and understandable way.

Stable means that a proposition testified today as true must be true tomorrow if the relevant preconditions are unchanged. If a customer states today that the requirements are completed, and if tomorrow he delivers further requirements or major change requests, we cannot attribute rationality to this organization. We suggest to take into consideration the following preconditions for ascribing rationality to an organization. The organization has defined regulations. In addition, by culture and tradition, or by management authority and power, these regulations are well implemented in the everyday work of the organization.

In other words, the other party does not need to have knowledge of the regulations and processes leading to a decision or a testimonial. The actor must at least produce similar testimonials under similar conditions. Knowing the conditions, another actor can then generate a prediction. Should this prediction fail repeatedly, it is difficult to ascribe rationality to the actor who makes a decision or testifies a proposition. Concerning the intentional stance according to Dennett (1981), we can describe how the actor ought to decide and what he ought to state if he is rational. From this, we can derive predictions about his rational behavior. This prediction can be derived from previous observations. Nevertheless, we will only be successful in our predictions if the actor acts under similar circumstances and in a similar way.

Against this background, it is remarkable that an SDP is an exception for many customers, whereas it should naturally be the supplier's daily business. In exceptional business cases, of course, the organization normally has no regulations implemented as a culture. We must face the fact that, in an SDP, a supplier will often be a rational agent, whereas the customer's maturity in decision-making in SDP may be lower. This corresponds to the fact that research literature in the field of project management and decision-making in SDPs focuses on the supplier's side. The awareness for research need is obviously higher than on the customer's side.

Therefore, we are only justified in considering SDP customers and SDP suppliers as rational actors under the stated precondition of well implemented relevant regulations on both sides. Only then, we will be entitled to use rationality-based concepts from economic theories and game theory for the description and understanding of the project parties' behavior.

4.2 Actors inside the SDP: Groups and Joint Teams

Inside the SDP, there are a lot of comparatively stable groups, but also temporary teams. We do not wish to make detailed empirical claims about the structure of these groups, but from the considerations of the previous section we will derive some criteria of attributing rationality and intentionality to these groups.

Some parts of an organization are stable groups. They act like organizations and have implemented regulations by authority or by culture. Development departments working according to quality standards as CMMI or using agile methods (see Zannier and Maurer, 2007) are rational actors as we see them. We can also maybe consider the customer's procurement department as a rational actor, because it follows documented regulations when selecting a supplier and negotiating the contract.

Other SDP stakeholders lack the criteria of intentionality and rationality in decision-making. If we consider the customer's business department in need of the software system, we can expect it to have implemented well-defined regulations in its everyday business. Therefore, it works as a rational actor. Nevertheless, it lacks those kind of regulations for requirement definition regarding the needed software, the testing of the delivered system and the testimonial production for its final acceptance.

Inside the SDP, the parties will often establish joint teams - some teams for project governance such as project management boards, and others for the processing of common tasks like specifying requirement details or performing system tests. We can hardly consider these teams as rational actors, but sometimes we can say that they perform joint actions. In SDP, we refer to this as collaboration. Collaboration is not necessarily in agreement with the rational goals of the organization (Axelrod, 1984; Ketchley, 2014). It is possible, in joint action, to see people from both parties agreeing to close testing before having executed all test cases, or agreeing to neglect security issues in the interest of saving time, or agreeing in reaching an agreed due date.

4.3 Individuals as Actors inside the Software Development Project

Finally, we have to consider the question if the individuals can be described as rational actors when working in an SDP. Since they are human beings, they have of course beliefs and desires. Furthermore, they have intentions and they will act accordingly. However, from the project's point of view, not all of these beliefs and desires are relevant. Researchers wish to describe the programmers, managers, analysts, and users as rational actors, using economic theories like game theory to understand their behavior (as in Yilmaz et al. 2010, Yilmaz and O'Connor 2012, Zannier and Maurer 2007). In this case, we must answer the question whether the involved subjects have desires and beliefs within the SDP context, and if their decision-making is rational with respect to these desires and beliefs. If they make their decisions in order to reach goals outside the SDP and not reducible to SDP relevant goals, or if they act strictly according to regulations or instructions from an authority, we are not justified in describing them as rational actors inside the SDP.

Table 2: Ontology of actors in software development projects.

Actor	Rationality of decision-making	How to understand the actor
Organizations and stable groups	Defined by regulations implemented by culture or authority	Understand the regulations, check the implementation degree
(Project) Teams	Mostly no rationality	Understand the members, check potential joint actions
Individuals	Human rationality, if not strongly guided by regulations of the organization	Understand individual beliefs and goals, check the implementation degree of regulations, check dependencies regarding communities and networks

Beside this, the behavior of individuals will influence the rationality of organizations and groups. The collective rationality of these actors depends on the rule-following of the involved individual people. Only if they follow the rules, based on authority and power of the organizations' managerial staff, or based on a well-implemented organizational culture, the organization will act rational. Since the individuals are not just embedded in one organization, conflicting demands from different rules may influence their behavior.

Furthermore, people may be part of one or more communities, having implemented their own culture. The culture of a community may determine the behavior of the individual also inside the SDP and can disturb the rationality of the groups and organizations.

Finally yet importantly, it is necessary to analyze the influence of implemented networks on the behavior of individuals. Networks may support the organization's defined norms and regulations, but they may also allow them to bypass the regulations. Networks then also disturb the organization's rationality.

5 CONCLUSIONS

In this paper, we introduced a social ontology in order to clearly differentiate between kinds of collective rationality as a basis of decisions, beliefs, and actions in the context of SDPs. We are thus contributing to clarify the social structure of SDPs. This will be a solid foundation for the application of economic concepts using the notion of rational actors to software development outsourcing and software project management research.

We have found three kinds of social actors inside the SDP. Table 2 sums up this classification.

First, there are organizations acting according to regulations well implemented by culture or authority. The involved companies may act as organizations of this kind, but also company departments or other stable, long-existing groups. We can ascribe rationality in decision-making to these organizations due to the understandable regulations they have implemented. Other actors can understand and predict these actors' behavior through the understanding of their goals and regulations if the implementation degree is at least observable from outside the organization.

Secondly, there are teams who just exist for a short time during the project. These teams often have no well-implemented regulations, and, therefore, we cannot ascribe rationality to these teams, although they may have goals and make decisions. It is difficult to understand and predict these teams, but we can use the concept of joint action in order to derive collective behavior predictions from the understanding of the members' individual beliefs and goals.

Finally, there are individuals. If they act according to their own beliefs and goals without a strong binding to regulations of the organization they belong to, we can ascribe a real human kind of rationality to them. We may understand and predict their behavior by using our own rationality. On the other hand, if they only act according to the organization culture or to authority instructions, they have no own rationality inside the SDP.

Based on an analysis of decision-making of collective and individual actors, it shall be possible to classify them as rational actors. This will be helpful to understand the stability of software requirements, conflicts between contractors, the dynamics of system usage, and software systems acceptance.

REFERENCES

Al-Ahmad, W., Al-Fagih, K., Khanfar, K., Alsamara, K., Abuleil, S., Abu-Salem, H. 2009. *A Taxonomy of an IT Project Failure: Root Causes.* International Management Review 5(1), 93-104.

Aubert, B.A., Patry, M. and Rivard, S. 2003. *A tale of two outsourcing contracts. An agency-theoretical perspective.* Wirtschaftsinformatik 45, 181-190.

Axelrod, R. 1984. *The Evolution of Cooperation.* New York, Basic Books.

Bakker, R. M., Knoben, J., De Vries, N., Oerlemans, L. A. 2011. *The nature and prevalence of inter-organizational project ventures: Evidence from a large scale field study in the Nether-lands 2006–2009.* International Journal of Project Management 29(6), 781-794.

Benaroch, M., Lichtenstein, Y., Wyss, S. 2012. *Contract Design Choices in IT Outsourcing: New Lessons from Software Development Outsourcing Contracts* (April 20, 2012). Available at SSRN: http://ssrn.com/abstract=2137174 or http://dx.doi.org/10.2139/ssrn.2137174.

Beulen, E. and Ribbers, P. 2003. *IT Outsourcing Contracts: Practical Implications of the Incomplete Contract Theory.* Proceedings of the 36th HICSS.

Dennett, D. C. 1981. *True believers: The intentional strategy and why it works.* In: Scientific Explanation. Ed. by Heath, A., Oxford University Press. 53-75.

Dwivedi, Y.K., Ravichandran, K., Williams, M.D., Miller, S., Lal,B., Antony, V., Muktha, K. 2013. *IS/IT Project Failures: A Review of the Extant Literature for Deriving a Taxonomy of Failure Factors.* IFIP Advances in Information and Communication Technology (402), 73-88.

Edum-Fotwe, F. T., Price, A. D. 2009. *A social ontology for appraising sustainability of construction projects and developments.* International Journal of Project Management, 27(4), 313-322.

El Emam, K., Koru, A. G. 2008. *A Replicated Survey of IT Software Project Failures.* IEEE Software 25(5), pp. 84-90.

Gaebert, C. 2014a. *Contract Design and Uncertainty in Software Development Projects.* Perspectives in Business Informatics Research. 217-230.

Gaebert, C.2014b. *Dilemma Structures between Contracting Parties in Software Development Projects.* Proceedings of the 9th International Conference on Software Engineering and Applications – 29-31 August 2014, SCITEPRESS – Science and Technology Publications, 539--548.

Gallotti, M. L., Pleasants, N., Griffiths, P. 2011. *Naturally We.* A Philosophical Study of Collective Intentionality http://hdl.handle.net/10036/2997.

Ketchley, N. 2014. *"The army and the people are one hand!" Fraternization and the 25th January Egyptian Revolution.* Comparative Studies in Society and History, 56(01), 155-186.

Kotter, J. P. (2008). *Corporate culture and performance*. Simon and Schuster.

Lawson, T. 2014. *A conception of social ontology.* In Pratten, S. (Ed.): Social Ontology and Modern Economics, 19-52.

Marschollek, D. K. F. O., Beck, R. 2012. *Alignment of Divergent Organizational Cultures in IT Public-Private Partnerships.* Business & Information Systems Engineering, 4(3), 153-162.

Narotzky, S. 2007. *The Project in the Model.* Current Anthropology, 48(3), 403-424.

Pettit, P., Schweikard, D. 2006. *Joint actions and group agents.* Philosophy of the Social Sciences, 36(1), 18-39.

Pratten, S. 2014. *Introduction.* In Pratten, S. (Ed.): Social Ontology and Modern Economics, 1-16.

Pratten, S. 2013. *Post-Keynesian Economics, Critical Realism, and Social Ontology.* The Oxford Handbook of Post-Keynesian Economics, Volume 2: Critiques and Methodology, 2, 62-739.

Searle, J. R. 2006. *Social ontology. Some basic principles.* Anthropological Theory, 6(1), 12-29.

Standish Group 2010. *CHAOS MANIFESTO, The Laws of Chaos and the CHAOS 100 Best PM Practices* The Standish Group International.

Tamburri, D. A., Lago, P., van Vliet, H.. 2012. *Organizational social structures for software engineering.* ACM Comput. Surv. 46(1), 1-34.

Tamburri, D. A., Lago, P., van Vliet, H., 2013 *Uncovering Latent Social Communities in Software Development.* IEEE Software, 30(1), 29-36.

Tollefsen, D. 2002. *Organizations as true believers.* Journal of social philosophy, 33(3), 395-410.

Tollefsen, D. 2004. *Collective epistemic agency.* Southwest Philosophy Review, 20(1), 55-66.

Tollefsen, D. 2011. *Groups as Rational Sources.* Collective Epistemology, 20, 11-22.

Yilmaz, M., O'Connor, R. V., Collins, J. 2010. *Improving software development process through economic mechanism design.* In: Systems, Software and Services Process Improve-ment. Ed. by Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R., Berlin Heidelberg: Springer. 177-188.

Yilmaz, M., O'Connor, R. V. 2012. *A market based approach for resolving resource constrained task allocation problems in a software development process.* In: Systems, Software and Services Process Improvement. Ed. by Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R., Berlin Heidelberg: Springer. 25-36.

Zannier, C., Maurer, F. 2007*: Comparing decision making in agile and non-agile software organizations.* In: Agile Processes in Software Engineering and Extreme Programming. Ed. by Abrahamsson, P., Marchesi, M., Maurer, F., Berlin Heidelberg: Springer. 1-8.