

# Communication in Agile Software Projects: Qualitative Analysis using Grounded Theory in System Dynamics

Seyed-Ali Marjaie, Urvashi Rathod

Symbiosis Centre for Information Technology (SCIT)

Plot 15, Rajiv Gandhi InfoTech Park, Hinjawadi, Pune - 411057. Maharashtra. India.

Tel: +91 020 2293 4308 / 4309 / 4310

Ali.marjaie@associates.scit.edu, Urvashi@scit.edu

**Abstract:** *Finding the difficulties and concerns of the client, understanding the variables involved in creating the difficulties, exploring the causality amongst the identified variables and framing the dynamic hypothesis is the most critical and difficult phase of system dynamics modeling process. Although, there are many ideas in literature to carry out this phase, lack of a comprehensive qualitative method is noticeable. Hence, the authors of current paper introduce an inclusive and customized grounded theory method, specific to requirements of qualitative aspect of system dynamics modeling. Additionally, this paper argues that the methodology described in the paper integrates existing methodologies to facilitate the iterative process of modeling. This work proposes a systematic framework in order to relate findings of qualitative phase to quantitative phase in system dynamics modeling process with a small prototype application in the real world problem.*

**Keywords:** *System Dynamics, Grounded Theory, Qualitative System Dynamics, Methodology, Agile Software Projects, Project Communication;*

## 1. Introduction

Almost a decade ago writers of *Manifesto for Agile Software Development* stated their objective of writing this manifesto as "...uncovering better ways of developing software..." through a shift in software development principles. Among other principles, agile manifesto shifted value system from appreciating processes and tools to more valuing "*Individuals and Interactions*". Also in new value system "*Customer Collaboration*" got more attention than contract negotiation and they emphasized on "*Responding to Change*" rather than following a plan (Beck et al., 2001). These three principles out of twelve can be considered as original sparkles of the idea for this research.

The term "Interaction" as stated in the Oxford dictionary expresses a reciprocal or mutual action. Human Interaction happens for two individuals or two groups of people in the presence of a suitable platform or context. Probably, the most important platform we can point out is a communication platform, which in turn ensures sufficient frequency of communication, inter and intra project team.

We believe that communication is an essential part in the process of collaboration, as well. The term “Collaboration” suggests act of working jointly. One cannot imagine the collaboration amongst individuals or teams without communication as a common facilitative platform. Taking into consideration, the amount of flexibility desired in agile manifesto, responsiveness to change, too, among other things needs an effective communication platform. Responding to change, in an indirect way depends on the same communication platform, which facilitates interaction and collaboration.

On the other hand, reciprocal or mutual nature of interaction, collaboration and responsiveness to change, a communication platform is insufficient without an appropriate *feedback mechanism*. A feedback mechanism ensures the mutual accountability of the participants for achieving the three basic values of agility in software development. Hence, we believe *communication and feedback mechanism* both are important parts of practicing agile methodology, worth studying in detail as they seem to be critical to the performance of agile projects.

According to the birth date of Manifesto for Agile Software Development (Year 2001), probably, the very first particular attentions to communication in agile software development goes back to not earlier than a decade ago. However, role of communication in agile manufacturing teams has been at researchers’ notice from long back (Forsythe, 1997). Also, there is no doubt that communication, per se, is as old as history, being one of human’s initial challenges. Although, this study is not the first attempt to bring up this subtle point in agile software development area, but the way we approach to understand and study communication and feedback mechanism in an agile environment is a unique and relevant approach.

This research particularly intends to study *Communication and Feedback Mechanism System* (henceforth, CFMS) and problems caused by its absence in agile software development projects. As mentioned earlier, almost a quarter of agile software development principles depend on CFMS. Therefore, not only we believe that CFMS is a sine qua non for an agile project, but also we contend that the functionality of the CFMS is critical to *success or failure* of agile projects. In our understanding a CFMS is a collection of entities and relationships, including individuals, who influence others by their behaviour or get influenced by others, while generating, receiving and demanding information in the desired way. In order to study CFMS as a whole or its outcome, we need to focus on constituent parts of it and study their attributes, behaviors, causes, and effects.

Also, while studying CFMS of an agile project, one comes across hundreds of constituent parts influencing each other in a desired or undesired way. Considering level of interconnectivity of all these constituent parts, the bounded rationality of human mind would limit the analysis at certain level and would fail to explain the functionality of the system effectively. Introducing time factor into mentioned system will contribute a considerable amount of dynamicity, which in turn, will increase complexity of the system further.

Research methodologies based on quantitative and statistical approaches provide a snapshot of current or past status of the field. When it comes to an experiment, which

studies a theory and expects immediate response of the system in laboratory environment (simulated situation), quantitative methods fall short to provide a dynamic solution. However, some methodologies and tools are useful in identification of root causes of complex systems (e.g., Ishikawa diagrams), but again they fail when time factor comes to picture (Ishikawa, 1990). We believe applying System Dynamics (SD) (Forrester, 1958) to complex communication systems in agile Information System (IS) projects is a hope to assure success of IS development projects, increase smoothness of the operation, and deliver business value (Abdel-Hamid and Madnick, 1989a; Madachy, 2008).

A blend of IS research with SD methodology creates the opportunity to use SD modeling to add a unique dynamic insight to one of the important challenges of agile environments - communication. In this study, we will take advantage of SD in order to model communication and feedback mechanism of an agile IS project. This research work tries to simulate interplay of communication and feedback mechanism in agile software development project. The main objective of the study is to model dynamics of communication and feedback system with regard to existing challenges to attain flexibility and responsiveness to fast-paced business needs. The contribution of this work can be appreciated for integration of selected qualitative approaches for facilitating the development of a causal structure aligned to the communication in agile projects and modeling its dynamic behavior. The authors envisage that this work is a beginning of a long term research study that aims to clarify the complexity of communication and feedback mechanism in agile IS development projects, and would contribute to the design of a flexible and effective communication system.

This paper has a section on the background of research addressing the literature on the adopted research approach and communication in agile project management. This section is followed by the discussion on research methodology, mainly, the Grounded Theory and qualitative aspects of System Dynamics. Detailed analysis of the data related to communication in agile project management follows the section of research methodology. The paper concludes after formulating some guidelines for the usage of analytical methods of Grounded Theory in System Dynamics.

## **2. Research Background**

### **2.1. Qualitative research methods and system dynamics**

In System Dynamics literature, many models discuss qualitative concepts with qualitative variables. Quality in life, quality and attitude, quality in corporate growth, and awareness of advertising are some examples of such concepts (Luna-Reyes et al. 2003). Sterman (2000) in his famous book of business dynamics dedicated a chapter to present a practical method for eliciting information about nonlinear relationships from experts. In this chapter, he proposes an interesting method of nonlinear function formulation based on qualitative and quantitative data and explains the process through different examples from manufacturing, the service industry, product development, and others. In one of the examples, he explains how Oliva (1996) incorporated interviews, archival data collection, and participant observation in order to test his model (Sterman, 2000). There are some more examples of modeling process, in which formulation is

grounded in the interviews conducted with employees of the organization, and illustrated with quotations and memos from those interviews in Sterman's (2000) work. Luna-Reyes et al. (2003) state that perhaps the richest set of examples of qualitative and judgmental modeling tools, and their incorporation in model formulations resides in the group model building literature ((Reagan-Cirincione et al. 1991; Vennix et al. 1992; Morecroft and Sterman 1994; Richardson and Andersen 1995; Vennix 1996; Andersen and Richardson 1997; Vennix et al. 1997)).

Despite the fact that different examples of incorporating qualitative data into model formulation elicited from mental models, interviews, written databases, and archival data, exist in literature, these methods are not always easy to implement and plain enough to make a standard inclusive process from them.

## **2.2. Communication in agile software project management**

Many researchers have made attempts to define communication in scientific words but establishing a single definition has proved impossible and may not be very fruitful (Carter, R. F., 1979). It has further emerged as a dynamic process that individuals use to exchange ideas, relate experiences, and share desires through speaking, writing, gestures or sign language (Glen & Smith, 1998). Management of communication in a project is considered as one of the important roles of a project manager. It is related to employing the processes required to ensure generation, collection, distribution, storage and retrieval of the project information in effective and timely manner (PMI, 2000). In traditional project management, usually project manager is at the hub of communication and all stakeholders involved in the project are connected through the communication plan designed by the project manager. However, as the complexity of projects increases and new types of project teams such as virtual, distributed and agile emerge, the traditional communication plan fails to fulfill communication needs of newly formed project teams. Although, teams may allocate different methods of communication in order to satisfy the communication requirements, the impact of adopting new communication approach remains unknown on productivity and quality of the product before experiencing it in reality.

The issue of software productivity and software quality haunts us perpetually with no single solution (Jain & Ting, 1989). Ambler (2005) considers agile quality to be a result of effective communication techniques, among other factors (e.g., effective collaborative work, incremental development, and iterative development as implemented through techniques such as refactoring, test-driven development, and modeling). As cited in Walston and Felix (1977), Boehm (1987), and Vosburgh, Curtis, Wolverton et al. (1984) found that the effect of tools is relatively small, whereas the impact of people and organization is significant on software productivity. Curtis et al. (1988) have studied software development using a layered behavioral model and have concluded that the design of software systems must be treated as learning, communication, and negotiating process.

In agile software development, not only generating and storing ideas, thoughts and self analysis is important, but also impartation of generated information and exchanging them has a significant role in success and failure of the project. In an Agile software development project, effective personal communication among team members and project participants has been highlighted by many so far (Abrahamsson, Salo, Ronkainen, & Warsta, 2003; Cockburn, 2003). Communication can happen in two main

forms, a serious cooperation in order to accomplish a task, or a discussion to solve some difficult problems (Wu & Sahraoui, 2005). We assume that in addition to these two forms, communication can be considered as sharing of experiences, achievements, and current status of a certain task or entire project. Special significance of communication in agile methodology probably comes from instability of requirements which demands and embraces personal communication between the participants (Abrahamsson et al., 2003). Instability of requirements, in turn, can increase ambiguity of mutual understanding process and personal communication is a natural reaction to high level of ambiguity in agile software development (Korkala et al. 2006). For example, conversation as a form of communication and social interaction, are considered as key building blocks of knowledge sharing in agile software development practice. Higher the levels of complexity or ambiguity, more the need for interactive knowledge sharing by means of direct verbal communication (Melnik & Maurer, 2004).

Many other factors play an intensifying role and makes communication more crucial. For example, by increasing the size of the team, communication becomes more challenging process. The communication remains manageable in small teams, and small teams suffer less from conflicting views. A quality product also results when the application knowledge is widely spread across the team, and with smaller teams it is easier to propagate this knowledge (Jain & Ting, 1989).

Tuomas Niinimäki and colleagues (2009) proposed a research framework to study communication in agile software projects. In addition to this research methodology they offered, a valuable list of key factors involved in communication. Forsythe (1997) compares an agile enterprise with a traditional enterprise based on certain factors such as collaborative work, parallel information flow, direct and indirect lines of communication, sequential information flow, negotiability of factors, number of changes in requirements, frequency of changes in requirements, and geographical separation. However, his research is not focused on software development and discusses agile manufacturing in general. Forsythe (1997) briefly summarized the human factors related to communications and information infrastructure required in a fully agile manufacturing environment, which influences an agile business practice.

Frequency of communication in distributed teams, compared to traditional project teams is higher (Galegher & Kraut 1994). Connaughton and Shuffler (2007) confirmed that frequency and face-to-face communication emerge critical consistently in research related to virtual teams. Hinds and Mortensen (2005) asserted that frequent communication enhances shared team identity and therefore, moderates the effect of distribution on interpersonal conflict. Through frequent communication team members are able to share their experiences and more effectively manage their incomplete and imperfect communication. Jarvenpaa, Knoll and Leidner (1998) and Jarvenpaa and Leidner (1999) in their respective research showed that frequent communication increases the trust in the teams. Also, they reveal that predictable communication with regular feedback has been associated with improved team performance ((Jarvenpaa & Leidner 1999); (Jarvenpaa, Knoll, & Leidner 1998); (Kayworth & Leidner 2000) and (Maznevski & Chudoba 2000)). On the other hand some research considered face-to-face communication as essential factor in order to promote trust (Oertig & Buegri 2006), lessen conflicts while doing a task (Hinds & Mortensen 2005), improve team dynamics, which, in turn, increases team effectiveness ((Maznevski & Chudoba 2000) and (Grosse 2002)).

To ensure the effectiveness of the communication process, means of communication such as speech, inscription, and symbols are the necessities to be considered (Mishra & Mishra, 2009). Type of tools depends on type of the communication taking place. For example, as cited in Mishra & Mishra (2009), according to Finsterwalder (2001), since, in face-to-face communication content of the communication can be forgotten after some time, tools such as whiteboards and electronic displays are used to store and retrieve the content of communication.

A list of factors extracted during literature review on communication and feedback mechanism in agile software development is presented in table 1.

Planned interpersonal interaction Unplanned interpersonal interaction Distance Trust Commitment Project performance (time, cost, quality) Project scope Creativity Pairing Physical proximity Location Frequency of face-to-face interaction Duration of face-to-face interaction Unplanned meetings Planned meetings Meetings and events Documents Memos and letters Figures and tables Schedules Presentation graphics Forms (surveys, progress reports) Project management materials News, reports and announcements Policies Requests for information Collaborative work General Information (e.g., phone lists)	Design and design analysis Brainstorming Person-to-Person Discourse Group planning (e.g., daily stand-ups) Communications requiring unequivocal Immediate confirmation or response Communications where a need exists Communications in which a spontaneous Urgent Communications Team building Requests for information Meeting duration Sprint length Job satisfaction Communication satisfaction Common communication language Local languages (in distributed projects) Organizational commitment Team satisfaction Individual performance Process satisfaction Daily stand-ups Media richness Frequency of communication Team size
--	--

**Table 1: Factors captured from communication and feedback mechanism in agile software development**

As shown in literature of communication in agile software development projects, communication has a significant role in success or failure of agile projects. Also, many other factors have intensifying or moderating role in increasing or decreasing the influence of communication on objectives of the project in terms of cost, time, and quality. This study mainly concerns about variability of these factors mentioned in literature as a dynamic communication system. On one hand, observing and identifying all these variables is important, on the other hand, understanding the variability of these factors and consequently realizing dynamic behavior of a communication system is a difficulty. The difficulty is to answer the questions such as: How much we should communicate. What to be communicated? Who should communicate to whom? How communication affects the project measures? How to maintain effective communication in distributed agile teams? How distance and communication are related? How variability of measures such as frequency of communication, duration of

communication, distance, duration and frequency of planned and unplanned communication, are contributing to the difficulties.

Agility, embedded in the 'response to change', remains a concern according to principles of agile manifesto. Maintaining agility itself is a difficulty and it may be lost if the communication factors and measures vary in an undesired way. Objective of this study is to address the ambiguity caused by the variability of mentioned factors in agile software development projects. It intends to build dynamic model in order to understand and capture the variability of factors involved in communication system and the challenges persist in agile software development projects, using a qualitative system dynamics approach.

This work limits the scope of the study into communication system used in requirements gathering, development, and testing phases of agile software development. However, we believe that among other things, defects management, bug fixing, quality of code, scheduling, velocity, sales and financial metrics and release management are important too, but studying and modeling mentioned areas is out of the capacity of the present work. Few main feedback structures related to quality and backlog in agile software projects are taken into consideration including human resource management and quality. This study tries to address at least most important pain areas of communication in agile software development projects relatively untouched so far by researches in a dynamic way.

### **3. Research Methodology**

Software is inherently complex (Brooks, 1987) and the process of development of this product is further complex for the reason of being the intellectual work. Communication of the complex intellectual work across the members of project team involves several difficulties like number of communication channels depending on the team size or cognitive differences in individuals' understanding about the work artifacts or choice and availability of communication media. Agility and distributed development teams add many more apparent and non-apparent difficulties to communication. As the distributed agile development is yet emerging, majority of learning is expected to come directly from the field. The study of phenomenon in reality involves huge set of interlinked factors and in the absence of prior knowledge developed in the field, it is impractical to attain objectivity in observation. Eventually, there emerges a need for the study of the perceptions of practitioners related to situations, issues and solutions. This calls for the interpretive approach to research, wherein, multiple realities exist owing to the subjective understanding of the people in the research setting and also, of the researcher, who is an integral part of the process (Carcary, 2009; Lee and Hubona, 2009). As purported by the interpretivists, the researcher in interpretive research has to derive meaning from the circumstances, the people involved and the broad relationships, which are identified based on the study of the details deep under the surface of apparent reality (Carcary, 2009).

Considering the scarcity of research work done in the area of communication in distributed agile projects, it is imperative to adopt interpretive approach to research and therefore, we end up choosing qualitative research methods for the proposed work.

Assessment of appropriateness of the qualitative methodology to the proposed work has been done by considering the research context and associated characteristics.

Software development process involves several intermingling elements like people, technology, machines, infrastructure and other resources for solving unique problems in the real world (Pall, 2000; Shaw, 1990). Continuous and frequent interactions among a large number of elements make it inappropriate to address the issues in bits and pieces. In general, dealing only with one part of a complex system gives us the event-driven view of the world and fails to address the comprehensive phenomenon of the system. We understand that a number of continuously interacting elements leads to a large number of possibilities and thus, increases the combinatorial complexity (Sterman, 2000) of the dynamic system of software development process. As the study aims to develop an initial understanding about the communication in distributed agile software development, which is a complex and dynamic system, we find the System Dynamics approach appropriate for the study. System Dynamics would not only help in mapping *the possible relationships among parts of the problem* (Williams, 2005) but also in understanding that how these relationships are changing, when tweaked in.

Although, system dynamics signifies mathematical modeling of the real world system by depicting interdependent variables in quantitative terms, need and importance of qualitative analysis aspect of System Dynamics has been discussed for decades in the literature (Forrester, 1994; Sterman, 2000) and (Forrester, 1975; Randers, 1980; Richardson and Pugh, 1981; Roberts *et al.*, 1983; Wolstenholme, 1990 in Luna-Reyes and Andersen, 2003). Initial stages of system dynamics modeling process like problem definition and system conceptualization require the collection and analysis of qualitative data from the field settings. Qualitative data constitutes the major input to the modeling process and therefore, there is an undeniable need for the usage of appropriate qualitative approaches for initiation of modeling. Luna-Reyes and Anderson (2003) have demonstrated the effective integration of qualitative methods at every stage of system dynamic modeling process and suggested a group of methods suitable to fulfill the information needs of modeling. As the present work addresses a relatively new field of communication in agile software development processes, employing some suitable qualitative approaches that effectively cater to the information needs of System Dynamics modeling becomes obvious.

For the present work, Grounded theory has been chosen as an appropriate method for identification of concepts across textual matter (Glaser and Strauss 1967; Eisenhardt 1989; Strauss and Corbin 1998; Sterman 2000; Luna-Reyes and Andersen, 2003) and establishing relationships between the concepts for dynamic modeling of the system. Outcome of Grounded theory approach employed for data gathering and analysis is used as an input to System Dynamics to develop dynamic hypotheses of the research concerns. The nature of both the methodologies, System Dynamics and Grounded Theory, makes entire process of research, iterative.

Grounded Theory has been identified as an effective approach in System Dynamics modeling in the works of a few authors (Rahmandad and Repenning, 2008; Luna-Reyes and Andersen, 2003). We customized general way of conducting qualitative research based on Grounded Theory to adjust the outputs with requirements of System Dynamic modeling. These small customizations created a unique methodology, which we found helpful in building dynamic hypotheses, based on qualitative analysis (of case study, interview, observation, etc.) for creating System Dynamics models. In the following sub-sections, we discuss both approaches that we combined for the present work.



### 3.1 Grounded Theory

Grounded theory propagates the idea of development of theory from the field-based observations and data. Continuous interactions with the field are fundamental to this approach for enhancing the depth and breadth of the underlying conceptual framework of the phenomenon under study. Theoretical concepts, thus developed, are easily understandable due to their origin in reality and are capable of representing the complex world in relatively effective manner (Turner, 1983). Such concepts and evolved theories remain available for criticism and corrections.

*Grounded Theory signifies an associated set of enquiry methods that provide flexible, successive analytic strategies for constructing inductive theories from the data* (Charmaz and Henwood, 2008). This approach has roots in the principles of continuous change and rejection of strict determinism and nondeterminism (Corbin and Strauss, 1990). It contends that there is a continuous change in the phenomenon due to evolution and therefore, methods must provide for accommodating the change. Also, it considers that actors' responses to situations depend on their perceptions, and as the conditions are changing continuously, there has to be a variation in their responses.

Considering the continuous evolution of actors perceptions and interpretations of situations in the real world, methods in Grounded Theory aim to uncover the underlying conditions, their interplay, actors' responses to those and consequences of those responses effectively (Corbin and Strauss, 1990). Data collection and analyses are interrelated in this approach. Data is gathered from interviews, published articles, account of observations or any other source from the research setting. These pieces of data are analyzed for finding out some clues or indications related to underlying phenomenon.

These clues raise more related questions for clarity and lead to next cycle of enquiry through interviews or observations. For example, members of a project team may be interviewed initially and a few responses like "We knew, Ram would be quick on deciding something that enhances customer satisfaction" and "I can easily debug the program written by other team members", would direct the researcher to frame further interview questions to explore the extent of knowledge of team members about each other.

Further comparison of data develops the indicators that convert the clues into conceptual labels, which are important for explaining the phenomenon. For example, more responses similar to the aforesaid statements would lead the researcher to the concept of "team's shared mental model". This process of comparing the events, incidents, observations, interview responses or any other relevant text for similarity and differences and assigning some relevant conceptual label to those is called *open coding*.

Indicators derived from the open coding process are strengthened or weakened by collection of new data and further analyses. These indicators (open codes) are compared for similarities and differences to discover the categories at higher level of abstraction. For example, while observing the team phenomenon, one can notice that concepts of team mental model, team collaboration and team experience are related to the higher level category, namely, team effectiveness. Thus, a category is a higher level concept

that integrates conditions, context, desired and actual actions and the consequences of actions in real world as visible in phenomenon.

Evolution of concepts, categories and subcategories involves several iterations for developing clear and right understanding and the notes related to each stage of developing understanding are written as memos. Memos are the text matter created for retaining the understanding attained at one stage, queries that emerge for achieving more clarity or any other idea relevant to the process of coding and analysis.

Critical analysis of relationships between categories leads to emergence of hypothesis and taken back to the field for assessing the need for revision. In this way, hypotheses are evaluated and modified for applicability in all evidences related to the phenomenon under study. When a hypothesis has been successfully evaluated and holds true for all possible situations related to the context of research, a theory grounded in real world data is built up.

It is easily noticeable that how Grounded theory is effective in capturing the comprehensive view of the real world phenomena. This approach creates the scope for evolution of theories with time and in accordance with the environmental conditions as perceived by the actors and researcher. Researchers gather data, compare it, try to make *all possible theoretical understanding* (Charmaz and Henwood, 2008) with appropriate reasoning and derive initial interpretations that further been verified and enhanced by going back to the field, collecting more data, developing better understanding and improving the reasons for the emerging interpretations.

The whole process of gathering subjective data from the research setting, decomposing it for comparison and finding out differences and similarities for open coding and evolution of categories and relationships signifies the strength of a disciplined approach of reasoning and verifying for the development of inductive theories.

Analytical methods of Grounded theory have their own weaknesses when implemented in practice and some authors (Corbin and Strauss, 1990; Elliot and Jordon, 2010) have given a detailed account of the ways of ensuring their sanctity for obtaining dependable result. For this study, we chose this approach for developing initial understanding of communication in agile projects by open coding. The next section discusses the system dynamics modeling process that carries forward the open codes identified in Grounded theory for hypothesizing and evaluating causal relationship between the variables relevant to the communication process.

### **3.2 System Dynamics**

Fundamentals of System Dynamics lie in the thinking that the world is a complex system of several interrelated things and the behavior of the system changes with time. System Dynamics is a perspective and a set of conceptual tools to enhance learning about dynamic complexity of systems around us (Sterman, 2000). These models are mathematical representation of the problem, developed by following a defined model building and evaluating process, which is iterative in nature (Luna-Reyes and Andersen, 2003). In these models, macro-level observable variables (Sawyer, 2007) are quantified and their relationships and interdependency is mapped to the numerical world for establishing causality and generating feedback theories.

Identification of macro-level observable variables needs analysis of qualitative data that resides in the mental models of the real-world actors relevant to the problem. Such qualitative analysis only leads to the initial formation of concepts related to variables and their relationships, further concretization of the same and extraction of numerical structure associated with the variables. Causal interdependence between the variables leads to the development of a feedback structure that captures real world patterns in space and time. The developed feedback structure explains the dynamics of the system in terms of a dynamic hypothesis *that represents a feedback theory or causal structure generating a series of behaviors over time, allowing the problem actors to learn about the situation, and to design or redesign their guidance policies* (Luna-Reyes and Andersen, 2003). Evolution of guidance policies, which influence the real-world problem, includes creation of *new strategies, structures and decision rules* (Sterman, 2000). The system dynamics model for the given problem provides the actors with a mechanism that facilitates experimentation, observation and further refinement. The process of model creation has been defined in literature in detail (Sterman, 2000; Luna-Reyes and Andersen, 2003) and may have three to seven steps. A seven step process is more descriptive and has the following steps:

**Step I: Problem Definition** – This step takes care of identification of problem, formation of important concepts related to the problem, recognition of the time frame, wherein, the concepts and problems are relevant and an initial proposition on the future behavior of concepts and variables based on the study of the past and the present (i.e., the reference mode).

**Step II: System Conceptualization** – In this step, existing principles governing the problem are analyzed, a dynamic hypothesis that explains the independent effect of feedback structure is formulated and causal structures based on the initial hypotheses, key variables and underlying interdependence are built.

**Step III: Model Formulation** – At this step, the modeler structure the model using variables, measurement scale, behavioral relationships, initial conditions and decision rules. This model simulates the real world problem under study.

**Step IV: Analysis of Model Behavior** – Model behavior has to be analyzed for assessing its ability to reproduce various modes of the system's behavior with the expected level of difficulty embedded in it. This step studies whether the changes in assumptions lead to anomalous behavior and whether the model exhibits surprise behavior in some novel conditions.

**Step V: Model Evaluation** – Model is evaluated for the consistency and correctness of its structure with respect to the system description, appropriateness of parameter values and for the way it handles extreme conditions. Model sensitivity to the changes in parameters is examined from the perspective of values, behaviour and policy.

**Step VI: Policy Analysis** – Consistent and comprehensive model is then used for framing the policy by anticipating the environmental conditions that may arrive, by examining the possibility of trying new decision rules and structure and by studying the effects of policies.

**Step VII: Model Use** – Policies emerging from the use of the model have to be evaluated for their robustness in uncertainties by changing the scenarios. Assessment of

resultant situations caused by the policy made using the System Dynamics model lead to the decision on policy implementation.

Every step of modeling process gets strong support from tools and modeling techniques of System Dynamics. Initial steps are quality data intensive and are supported by several relevant methods. Mashayekhi and Ghili (2010) have given a detailed account of usage of qualitative analysis for problem definition. In their work, the authors have established that the difficulties in System Dynamics modeling have their roots in ambiguities related to phenomenon. Researchers can identify the conceptual variables easily by addressing the ambiguities that have associated difficulties.

For the initiation of the present work related to the communication in distributed agile projects, we decided to use a combination of the stated approaches. For identification of difficulties in defining the problem, Coding approach of Grounded Theory has been used. Relevant questions related to the difficulties helped in extracting the ambiguities, which are in the root of the difficulties. These ambiguities have been instrumental in identification of macro-level conceptual variables for the modeling. The present work is limited to the identification of variables and step by step account of the combination of methods.

This is a novel approach and will definitely have some limitations that would emerge only after model formulation, testing, policy making and evaluation. We are proposing this approach only as a proof of concept developed through deliberations on literature and knowledge gathered from the practice.

## **4. Analysis and findings**

### **4.1. Sources of data**

Main source of information for modeling process is identified as qualitative data (Forrester 1975; Randers 1980; Richardson and Pugh 1981; Roberts et al. 1983; Wolstenholme 1990 in Luna-Reyes and Andersen, 2003; Sterman 2000).

In this study, as a main source of data, we use interviews conducted by InfoQ organization from agile experts who have had a participation in Agile Tour conference in the years 2009 and 2010 (InfoQ 2011). Selected number of interviews collected based on relevance to the concept of communication in agile software development approach. Interviews are specifically focused on difficulties and concerns of field experts, scrum masters, project managers, authors and researchers in agile software development.

We also referred to literature in order to find the existing major concerns in the field and to study the steps, already taken, for solving the communication related problem in agile environments. Literature, mainly, on comparison between traditional methods of software development and agile approach, led to many relevant factors.

### **4.2. Codification - Defining Difficulty**

As discussed, Grounded Theory (Glaser and Strauss 1967; Eisenhardt 1989; Strauss and Corbin 1998; Sterman 2000) is a method of exploring the data sources and coding them according to the purpose of the study. From the point of view of the methodology, codes serve a variety of purposes. They capture meaning in the data. They are used to identify the focus areas of the study. Also, they serve as handles for specific occurrences in the data. Codes are used as classification tags at different levels of abstraction in order to create sets of related information units (variables) for the purpose of

comparison (e.g., "communication strategy"). From an operational (low) level perspective, codes are typically short pieces of text referencing other pieces of text. The purpose in coding is to classify and deduct an often large number of sources including interviews, literature, case studies, etc.

In our methodology, we defined a new role for codes based on "difficulty" concept, which first introduced to System Dynamics modeling by Mashayekhi and Gilli (2010). As stated in their study, "difficulty" is nothing but a bothersome situation that needs to be addressed and resolved. During course of coding we used codes as "difficulty identifier" and "variable detector". In fact, this methodology considers "difficulty" identification as the starting point in System Dynamics modeling and uses coding to explore the sources of study in order to extract the some bothering points of the case.

The process started with listening to the interviews and coding relevant concepts and related topics which were relevant from the point of view of the agile software development difficulties and challenges related to communication. Main purpose of this phase of coding was to identify the challenges and difficulties, which practitioners of agile software development experience in the field.

In first attempt of identifying difficulties, unintentionally, we coded all facilitators and tools for communication. We extracted a wrong layer of codes. Since, we were looking for difficulties in communication; we realized that by removing facilitators of communication from the context, difficulties will emerge. In second attempt, we extracted all difficulties mentioned by interviewees during interviews.

The approach for coding the difficulties is a simple procedure. Some of the difficulties were mentioned directly by the interviewee as a challenge to project communication. The rest, we identified by hypothetically removing the tools and media of communication from the context, and monitoring the difficulties emerged from the hypothetical situation. Orientation of questions is based on flow of conversation and sometimes it goes off the topic. However, we tried to grasp the core concepts conveyed by interviewees related to the research concern. As suggested in the methodology of Grounded Theory (Glaser and Strauss, 1967) we followed theoretical sampling approach in data collection. We strongly avoided collecting ample of sources at the beginning and started with just a general idea of the problem, analyzed and coded one related interview, and accordingly selected the next source of the study. In fact, as we coded the interviews, codes directed us towards next focus area. Theoretical sampling (Glaser and Strauss, 1967), in fact, is the course of collecting data in order to facilitate evolution of theory. During this process the analyst simultaneously collects data, codes, and analyzes the data to decide what should be the next focus area and what piece of data needs to be collected further, in order to develop the dynamic theory as it emerges (Glaser and Strauss 1967; Eisenhardt 1989; Strauss and Corbin 1998; Stermann 2000).

#### **4.2.1. Open coding**

Open coding process is useful in identification of important focus areas, concern, difficulties and variables. From the interview transcripts, codes were created as per the understanding of researchers. The codes should represent the subject. For example, we used "diff." prefix for difficulties and "var." for variables as naming convention. As coding proceeds, new codes were generated and assigned, also, in some cases repetitive subjects emerged, for which we did not create a new code but assigned already-

generated codes. The number of times a code gets assigned in the text is called groundedness of that code. Groundedness for a code is the number of pieces of data connected to certain code. Larger the groundedness, stronger the evidence, which already found in the case (Glaser and Strauss, 1967). In fact, groundedness is the number of quotations associated with certain code.

#### **4.2.2. Axial Coding**

Coding process after a while comes to a certain saturation point. Saturation means no new code can be identified and existing codes get reused as one progresses. At this stage, one stops the open coding process and enters into axial coding phase. Axial coding, in fact, is a way to categorize the generated codes. In this phase, researchers realize certain relationships between codes. We relate them to each other and create a network of related difficulties or variables.

At this stage, the initial dynamic hypothesis starts emerging considering how we relate codes or how we categorize them. Relationships are just to find the density of the certain codes. Density of a code is defined as the number of codes connected to certain code. Larger density of a code can be interpreted as a high degree of theoretical density (Glaser and Strauss, 1967).

#### **4.2.3. Memoing**

Memos are capture of thoughts, dynamic hypothesis and our notes during coding process. A memo may stand alone or it may refer to quotations, codes, and other memos. They can be grouped according to types of memos, which is helpful in organizing and sorting codes. In fact, memos are containers of dynamic theories one may come across or one may generate. We used memos in order to note down all the important ideas passing through mind through coding process and finally framing ambiguity statements.

Mashayekhi and Gilli (2010) introduced a concept of ambiguity to System Dynamics problem definition. They define ambiguity as “a puzzle or a question that its answer may help solve the difficulty”. Ambiguity is nothing but a statement carefully composed in order to address unseen aspects of a difficulty. In fact, the process of phrasing ambiguity statement and attempts to uncover difficulties behind the ambiguity is a way leading towards the core and root cause of the problem.

The process of creating ambiguity statements with the help of questions asked by “what..?” and “why..?” is an iterative process. An ambiguity becomes more effective and clearer if deeper questions find deeper causes or reasons for the difficulty. During this iterative process, sometimes, new variables emerge, which can change the results of the analysis and modeling. This iterative process of creating ambiguity statements continues till a difficulty is solved. In fact, any unsolved difficulty has at least one ambiguity in its root, based on that we can build up a problem definition.

We used the same process to identify ambiguity statements and identified problems by memoing. Outcome of this process was a difficulty with a chain of ambiguity statements and a set of questions asked to clarify the difficulty. Meanwhile, during this question and answer, sometimes, we noticed new variables, which are involved in the system. We appended them into the variable-log.

### 4.3. Framing ambiguity statement

Iterations in above procedure ensure that you reach as deep as possible to the causes. As quoted by Mashayekhi and Gilli (2010) from Forrester (1969), “the root of a problem is not necessarily close to its symptoms in time and space”. Therefore, having multiple iterations can go beyond the symptoms and reach to the root of the problem. As iterations spin in, the seeker comes to know more about the root causes of the difficulty by asking more effective and strong questions. As the researcher proceeds to ask questions, more important variables will emerge.

Asking effective and related questions and phrasing ambiguity statements in this stage is an art. By asking different set of question one may end up collecting different set of variables, which, finally, will lead you to completely two different models. Therefore, we tried to keep evaluating questions and asking most effective and relevant ones during this process. However, we can't say that since two models are different one of them must be wrong. Forrester (1969) says “All models are wrong”. But the model, which addresses the identified difficulty in a comprehensive manner and let the researcher experiment with dynamic hypotheses, is the desired one.

Mashayekhi and Gilli (2010) suggest that one should keep iterating till the following conditions are met.

1. “We reach an ambiguity not answerable without modeling and simulating”;
2. “And we are prepared enough to build a model which can answer this ambiguity (or at least, can help us with the answer)”.

When the researcher reaches the above condition, she already has a dynamic hypothesis, she has variables involved in the system, and she has some relations emerged amongst the variables. At this stage, one starts modeling the existing scenario. After having the initial model ready, one can start experimenting based on the model, questions we have in our mind and the dynamic hypothesis made during coding and memoing.

### 4.4. Variable identification

List of the variables emerged from the discussions on ambiguity statements listed for the difficulties. While identifying the variable, its relevance to other variables or to the difficulty emerged, as well.

For example ambiguity statements of difficulty “proximity” is as below:

*“What makes proximity a difficulty? Distributed nature of software development. Why we need distributed software development approach? Because of amount of desired effort and expertise from different teams at a time and because of currency factor to have project done with minimum cost. What makes outsourcing difficult then? Finding an effective communication method makes it difficult. What makes a communication an effective communication? Richness of the medium and availability of the communication medium. What if we find an effective way to communicate? Then we can rely on distant communication. And we will find outsourcing possible.”*

Variables identified from above ambiguity statements and answers:

*“Desired effort, currency factor, outsourcing, effective communication, Richness of the medium, availability of the communication medium, rely on distant”*

Relationships between these variables can be shown as links in dynamic models.

## 5. Some guidelines for the usage of Grounded Theory in System Dynamics

Sterman (2000) suggests five steps for modeling process. Out of five first two steps requires a qualitative approach. We used grounded theory for first two steps. In this section, we point out some guidelines in order to use grounded theory based on those steps. In the box below we copy first two steps mentioned by Sterman (2000), and under each section we add our guidelines in order to incorporate grounded theory. Each guideline is supported by an example from the present study.

### Step 1. Problem Articulation

**Theme selection:** What is the problem? Why is it a problem?

Read the interview transcripts or literature. As a clue for the difficulty is found, or any symptom of the difficulty (that can lead to its source by asking related questions) appears, code the sentence or paragraph with an appropriate label. Name the code with a prefix like “diff.” and add a brief name as postfix. Prefix at code name latter would be helpful in generating list of certain type of codes.

Extracted cause(s) of the problem has to be coded as well. As you notice piece of data as a cause for a problem you should code it with “cause.” prefix. But the question is how to relate the cause into the problem? This relation is maintained by “relate code” facility of qualitative data analysis software (e.g., Atlas.ti). After relating two codes, one can retrieve them in hierarchy of causes. In some cases, one may guess the cause of the problem, and confirm it by asking confirmatory questions from the interviewee in order to check the validity of one’s guess. In this case, one can attach a memo to the code and explain in few lines the cause which one guess. Examples of codes, their relations and hierarchy of codes are shown in Fig. 1.

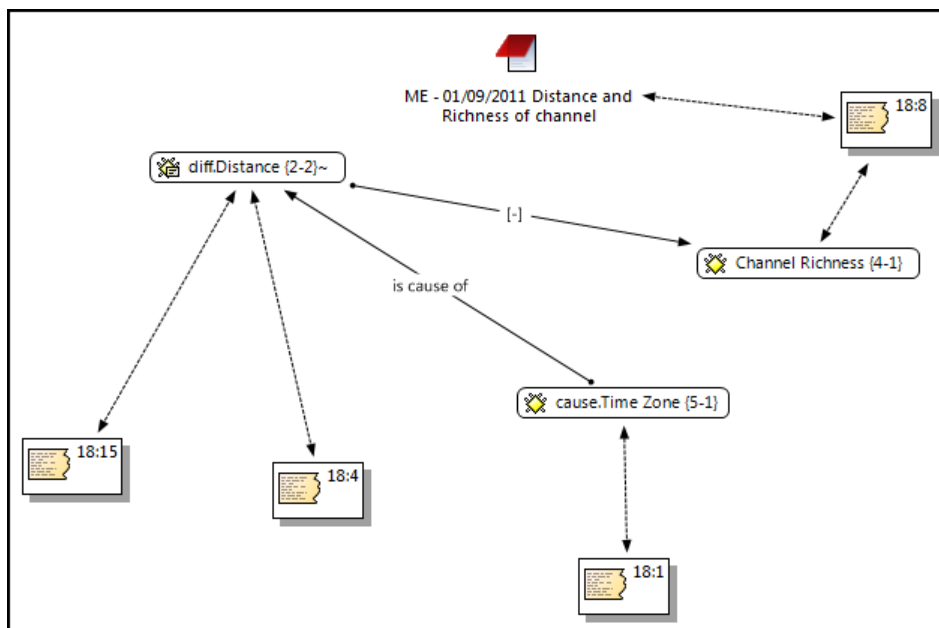


Figure 1. Illustration of a network of codes and memos; hypothesis generation and current theories of the problematic behavior



**Key variables:** What are the key variables and concepts we must consider?

While reading interview transcripts and/or literature, more variables emerge. Each variable and the context have to be coded. The prefix for variable coding can be “var.”. In this stage, as different types of variables emerge we can create three categories of endogenous, exogenous and excluded indigenous (Serman, 2000) variables. This simplifies the access to each type of variables during modeling process. An example of a category for endogenous variables with its variables is shown in Fig. 2.

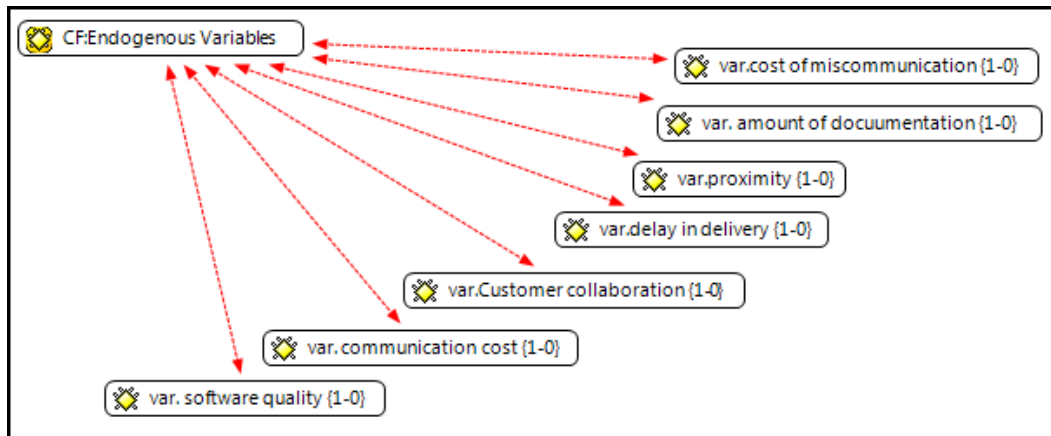


Figure 2. A category of variables

**Time horizon:** How far in the future should we consider? How far back in the past lie the roots of the problem?

**Dynamic problem definition (reference modes):** What is the historical behavior of the key concepts and variables? What might their behavior be in the future?

The selection of time horizon can severely influence the evaluation of policies (Serman, 2000). While deciding about the time horizon of the model, we should consider history of the problem domain to illustrate how it emerged, and describe its symptoms. Also, it should extend into the future to capture the delayed and indirect effects of potential policies (Serman, 2000).

We found Grounded Theory very helpful in understanding and finding the root and emergence era of the problem. While codification, you should code for symptoms of the problem in a chronological order with an intention to identify the suitable horizon for the emergence of those symptoms. We found codes in chronological order very helpful in identifying important spots of the reference mode, the key concepts and their variability during span of time.

## Step 2. Formulation of Dynamic Hypothesis

**Initial hypothesis generation:** What are current theories of the problematic behavior?

**Endogenous focus:** Formulate a dynamic hypothesis that explains the dynamics as endogenous consequences of the feedback structure.

**Mapping:** Develop maps of causal structure based on initial hypotheses, key variables, reference modes, and other available data, using tools such as Model boundary diagrams, Subsystem diagrams, Causal loop diagrams, Stock and flow maps, Policy structure diagrams, other facilitation tools.

Formulation of dynamic hypothesis using Grounded Theory is through memoing. Codification process goes hand in hand with memoing. We used memoing for recording our thoughts, identified causes, relations between variables and initial hypothesis generation. As shown in Fig.1, we used network structure of codes, memos and relations between them in order to illustrate a structured diagram.

## **6. Conclusion**

This study is an initiation of the process of understanding the issues related to the communication in distributed agile projects. For this initiation, a unique combination of qualitative methods has been used for achieving effective rigor in defining problem through difficulties, and identifying the variables that play important role in model formulation. A mix of analytical methods of Grounded theory and variable identification method using difficulties and associated ambiguities have been found to be useful for developing System Dynamics models. The work has been limited to the variable identification and will be carried forward for policy making using dynamic models.

## References:

- Abdel-Hamid, T. K. and Madnick, S. (1989), Software productivity: Potential, actual, and perceived. *System Dynamics Review*, 5: 93–113. doi: 10.1002/sdr.4260050202.
- Abdel-Hamid, T. K., & Madnick, S. E. (1989). Lessons Learned from Modeling the Dynamics of Software Development. *Communications of the ACM*, 32(12), 1426-1 438.
- Abrahamsson, P. P., Warsta, J. J., Siponen, M. T., & Ronkainen, J. J. (2003). New directions on agile methods: a comparative analysis. IEEE 25th International Conference on Software Engineering. Portland, OR, USA, 20030503.
- Ambler, S. (2005). Quality in an agile world. *Software Quality Professional*, 7(4), 34–40.
- Brooks Frederick P. (1987), No silver bullet: essence and accidents of software engineering, *Computer*, Vol. 20, Issue 4, pp.10-19.
- Carcary, M. (2009), The Research Audit Trial – Enhancing Trustworthiness in Qualitative Inquiry.” *The Electronic Journal of Business Research Methods* Volume 7 Issue 1, pp.11 – 24
- Carter, R. F. (1979). Theories of Human Communication (Book). *Journalism Quarterly*, 56(2), 420-421.
- Cockburn, A. A. (2007). Being in the room: lessons learned in collaboration. *Cutter IT Journal*, 20(8), 20-23.
- Cockburn, A., Highsmith, J., & Boehm, B. (2001). Agile Software Development: The People Factor. *Computer*, 34(11), 131.
- Connaughton, S. L., & Shuffler, M. (2007). Multinational and Multicultural Distributed Teams. *Small Group Research*, 38(3), 387-412.
- Corbin, J., & Strauss, A. (1990). Grounded Theory Research: Procedures, Canons, and Evaluative Criteria. *Qualitative Sociology*, 13(1), 3.
- 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.
- Elliott N. and Jordan J. (2010), Practical strategies to avoid the pitfalls in grounded theory research, *NURSERESEARCHER*, 17, 4, 19-40
- Forrester JW. (1958). Industrial dynamics: a major breakthrough for decision makers. *Harvard Business Review* 36(4): 37–66.

Forrester JW. (1975), Industrial dynamics—a Response to Ansoff and Slevin, in Collected Papers of Jay W. Forrester. Wright-Allen Press: Cambridge, MA, 151–165.

Forrester JW. (1994), Policies, decisions and information sources for modeling, *European Journal of Operational Research* 59(1): 42–63.

Forrester, J. W. (1968). *Industrial Dynamics*. Boston MA: MIT Press.

Forrester, J. W. (1969). *Urban Dynamics*. Boston: MIT Press.

Forrester, J. W. (1971). *World Dynamics*. Boston, MA: MIT Press.

Forrester, J. W. (1973). *Principles of Systems*. Boston: Wright-Allen Press.

Forrester, J. W. (1991). System Dynamics and the Lessons of 35 Years. In *The Systemic Basis for Policy Making in the 1990s*. Boston.

Forrester, J. W. (1994). System Dynamics, Systems Thinking, and Soft OR. *SystemDynamics Review*.

Forrester, J. W., & Senge, P. (1890). Tests for Building Confidence. *TIMS Studies in Management Science*, 209-228.

Forsythe, C. (1997). Human factors in agile manufacturing: A brief overview with emphasis on communications and information infrastructure. *Human Factors & Ergonomics in Manufacturing*, 7(1), 3-10.

Galegher, J., & Kraut, R. E. (1994). Computer-mediated Communication for Intellectual Teamwork: An Experiment in Group Writing. *Information Systems Research*, 5(2), 110-138.

Glaser, B. G. and A. L. Strauss. (1967), *The discovery of grounded theory; strategies for qualitative research.*, Aldine Pub. Co.

Grosse, C. (2002). Managing Communication within Virtual Intercultural Teams. *Business Communication Quarterly*, 65(4), 22-38.

Hinds, P. J., & Mortensen, M. (2005). Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science*, 16(3), 290-307.  
doi:10.1287/orsc.1050.0122

Hinds, P. J., & Mortensen, M. (2005). Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science*, 16(3), 290-307.  
doi:10.1287/orsc.1050.0122

InfoQ. (2011). "Agile Community Content on InfoQ." Retrieved February 1, 2011, from <http://www.infoq.com/agile>.

Jain, A. K., & Ting, P. D. (1989). Software quality via rapid prototyping. doi:10.1109/GLOCOM.1989.64048

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and Trust in Global Virtual Teams. *Organization Science*, 10(6), 791-815.

Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). Is Anybody Out There?. *Journal of Management Information Systems*, 14(4), 29-64.

Kathy Charmaz & Karen Henwood, (2008). Grounded Theory in The SAGE Handbook of Qualitative Research in Psychology. Editors: Carla, W., and Wendy, S., London: SAGE Publications Ltd

Kayworth, T., & Leidner, D. (2000). The Global Virtual Manager: A Prescription for Success. *European Management Journal*, 18(2), 183.

Korkala, M. M., Abrahamsson, P. P., Kyllonen, P. P., Chao, J. J., Cohn, M. M., Maurer, F. F., & ... Shore, J. J. (2006). A case study on the impact of customer communication on defects in agile software development.

Lee A. and Hubona G. (2009), A Scientific Basis for Rigor In Information Systems Research, *MIS Quarterly* Vol. 33 No. 2, pp. 237-262/June.

Luna-Reyes L. and Andersen D.(2003), Collecting and analyzing qualitative data for system dynamics: methods and models, *Syst. Dyn. Rev.* 19, 271–296

Margaret, O., & Thomas, B. (2006). The challenges of managing cross-cultural virtual project teams. *Team Performance Management*, 12(1/2), 23-30.

Mary S. (1990), Prospects for an engineering discipline of software, *IEEE Software*, Vol. 7, No. 11, November, pp. 15 – 24

Mashayekhi, A. N. (1992). From a Static Picture to a Dynamic Problem Definition. *International System*

Mashayekhi, A. N., & Ghili, S. (2010). Real Estate Cycles: A Theory Based on Stock Flow Structure of Durable Goods Markets. *Working Paper*.

Mashayekhi, A. N., Foroughi, H., & Hadavandi, M. (2006). Supply Demand World (SD W): An Interactive Learning Environment for Teaching Microeconomics. International System Dynamics Conference. Nijmegen, Netherlands: System Dynamics Society.

Maznevski, M. L., & Chudoba, K. M. (2000). Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness. *Organization Science*, 11(5), 473-492.

Melnik, G. G., & Maurer, F. F. (2004). Direct verbal communication as a catalyst of agile knowledge sharing.

Mishra, D., & Mishra, A. (2009). Effective communication, collaboration, and coordination in eXtreme Programming: Human-centric perspective in a small organization. *Human Factors & Ergonomics in Manufacturing*, 19(5), 438-456. doi:10.1002/hfm.20164

Niinimäki, T. T., Piri, A. A., & Lassenius, C. C. (2009). Factors affecting audio and text-based communication media choice in global software development projects. doi:10.1109/ICGSE.2009.23

Oliva R. (1996), A Dynamic Theory of Service Delivery: Implications for Managing Service Quality. Unpublished PhD Dissertation, Sloan School of Management, Massachusetts Institute of Technology. Cambridge, MA.

Pall Gabriel A. (2000), Process-Centered Enterprise: The Power of Commitments, St. Lucie Press, Florida

Rahmandad, H. and Hu, K. (2010), Modeling the rework cycle: capturing multiple defects per task. *System Dynamics Review*, 26: 291–315. doi: 10.1002/sdr.435

Rahmandad, H. and Reppenning, N. (2008), The Dynamics of Capability Development and Erosion. Appendix 3-1- Reflections on the data collection and modeling process., available at <http://filebox.vt.edu/users/hazhir/www/papers/ResearchProcess-June24.pdf> as last seen on March 21, 2011.

Randers J. (1980), Guidelines for model conceptualization, in Elements of the System Dynamics Method, Randers J (ed.). MIT Press: Cambridge, MA, 117–139.

Richardson G. and Pugh A. (1981), Introduction to System Dynamics Modeling with DYNAMO. Cambridge, MA: Productivity Press.

Roberts N., Andersen D., Deal R., Grant M. and Shaffer W. (1983), Introduction to Computer Simulation: The System Dynamics Modeling Approach. Addison-Wesley Reading, MA.

Sawyer K. (2007), Simulating Complexity in The SAGE Handbook of Social Science Methodology. Editors: William, O., and Stephen, P. T., London: SAGE Publications Ltd  
Sternan, J. D. (2000). *Business Dynamics, Systems thinking and Modeling for the Complex World*. Boston: McGrawhill.

Turner, B. A. (1983). THE USE OF GROUNDED THEORY FOR THE QUALITATIVE ANALYSIS OF ORGANIZATIONAL BEHAVIOR. *Journal of Management Studies*, 20(3), 333-348.

Walston, C. E., & Felix, C. P. (1977). A method of programming measurement and estimation. *IBM Systems Journal*, 16(1), 54-73.

Williams B. (2005), Systems and Systems Thinking, in Encyclopedia of Evaluation Sandra Mathison (ed.), SAGE Publications, Inc, 406-413.

Williams, L., & Cockburn, A. (2003). Agile Software Development: It's about Feedback and Change. *Computer*, 36(6), 39.

Wolstenholme E. (1990), *System Enquiry: A System Dynamics Approach*. Wiley: Chichester.

Wu, L. L., & Sahraoui, H. H. (2005). Accommodating software development collaboration.