What Constitutes Software Development Agility?

Emergent Research Forum Papers

Shekhar Rathor
Florida International University, Miami
srath004@fiu.edu

Dinesh Batra
Florida International University, Miami
batra@fiu.edu

Weidong Xia
Florida International University, Miami
xiaw@fiu.edu

Mingyu Zhang
Florida International University, Miami
mzhan004@fiu.edu

Abstract

Software development agility is the capability to manage various kinds of changes during the development process. Agile methods purport to facilitate processes that can address agility. However, the underlying dimensions of agility are not well elucidated in the literature. Specifically, what constitutes software development agility and what are organizational antecedents and outcomes of the agility dimensions are research questions that remain to be adequately answered. To bridge this literature gap, this study presents agility dimensions with corresponding descriptions that can help develop measures of agility. This study will contribute to the theoretical literature by developing better understanding about the measurement and nature of agility. Furthermore, the study will provide guidance to practitioners regarding specific processes for achieving agility in the agile software development process.

Keywords

Software development agility, agile software development, agility, agile methodology, delivery capability

Introduction

The need for adapting to changing customer requirements has resulted in shifting from plan-driven traditional software development methodologies to incremental and iterative development methodologies such as agile software development methodologies (Nerur et al. 2005). According to the 2015 Chaos report by the Standish group, agile software development projects are often three times more successful than projects based on traditional methodologies (StandishGroup 2015). Agile methodologies can enable organizations to achieve flexibility in software development process for managing unpredictable and changing conditions (Maruping et al. 2009). Although many studies have examined the theoretical and practical aspects of agile methodologies, agility still remains a nebulous concept, lacking in clarity, particularly about its underlying dimensions (Balijepally et al. 2014; Wendler 2013). There is a need for quantitative research to create empirical measures for software development agility (Conboy 2009; Sheffield and Lemétayer 2013). This research aims to fill this literature gap by empirically identifying key dimensions of agility and proposing their corresponding measures.

Literature Review and Conceptual Development

Agility and Team Delivery Capability

In software development literature, agility has been conceptualized in many different ways (Cockburn 2006; Conboy 2009; Highsmith 2004; Lee and Xia 2010; Lyytinen and Rose 2006; Sarker and Sarker 2009; Sheffield and Lemétayer 2013). In this study, we adopt Conboy's (2009) definition of agility: "the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and

simplicity) through its collective components and relationships with its environment" (p. 340). Based on the literature and our qualitative study on agility facilitators (Batra et al. 2016), we conceptualize three underlying dimensions of agility: sensing changes (Conboy 2009; Li et al. 2010), responding to changes (Conboy 2009; Lyytinen and Rose 2006) and learning from changes (Conboy 2009; Lyytinen and Rose 2006). Sensing changes refers to being proactive and anticipating changes so that the development process can become more effective and efficient (Conboy 2009; Li et al. 2010). The team should have the relevant business and technical skills in order to achieve anticipation capabilities, which in turn requires IT and Business teams to continuously collaborate. Responding to changes is an important constituent of agility because IT and business teams should be able to implement changes during the development process. Teams can achieve agility when they are able to respond to changes in an effective and efficient manner (Li et al. 2010). Agile principles emphasize the adjustment of team behaviors at regular intervals to become more effective and efficient (Conboy 2009; Highsmith 2004). For example, the Scrum method recommends for a retrospective meeting at the end of each iteration for the purpose of learning from project experiences (Chau and Maurer 2004; Schwaber and Sutherland 2011). Learning from changes is required in order to enhance over time team skills and project agility. Thus, agility can be conceptualized as consisting of sensing, responding, and learning cycles. Table 1 shows the three agility dimensions, their definitions and literature sources.

Dimensions	Definitions	Key References
Sense Change	IT and Business teams are able to anticipate various changes during the project	(Conboy 2009), (Li et al. 2010)
Respond to change	IT and Business teams are able to deal with various changes during the project	(Lyytinen and Rose 2006), (Conboy 2009)
Learn from Change	IT and Business teams are able to enhance their capabilities to sense and respond to various changes due to their past work experiences during the project	(Lyytinen and Rose 2006), (Conboy 2009)

Table 1. Agility Dimensions and their definitions

Team delivery capability refers to the team's ability to effectively and efficiently utilize member's skills, including technical, business, interpersonal, problem solving, and management skills, to successfully implement customer requirements. It is the capability of a team to deliver based on planned tasks. Team delivery capability is one of the critical success factors of agile software development projects (Chow and Cao 2008). We distinguish between agility and team delivery capability such that team delivery capability is the ability to deliver planned task and agility is the ability to deal with changes. If there are changes during the project, the team needs agility to deal with various types of changes.

Project Outcomes

Project success depends on the extent to which the customers perceive the project outcomes to meet or exceed their expectations. Project success can be defined in either objective (time, cost) or subjective (quality, customer satisfaction) measures (Siau et al. 2010). Customer satisfaction is one of the main emphases of agile values and principles. Agile methods increase customer satisfaction by frequently delivering value (Fontana et al. 2014), producing high quality products (Maruping et al. 2009), and dealing with frequent project changes. In this study, we conceptualize project success in terms of project efficiency parameters (time, cost) (Chow and Cao 2008; Lee and Xia 2010), customer satisfaction with changes and customer overall project satisfaction (Sheffield and Lemétayer 2013). It is important to distinguish between customer satisfaction from changes and overall customer satisfaction. Customer satisfaction from changes is a better indicator of project agility because it reflects customer satisfaction from various type of changes only. A team may be effective and efficient in doing planned tasks, but it may not do equally well in dealing with changes.

Aspects of Changes

Change is a central concept of agility (Conboy 2009). In order to define the context in which agility is needed, it's important to define and categorize the various kinds of changes that the software development process encounters. Agile methods such as scrum emphasize on responding to user requirement changes

only, but there can be other changes such as changes in hardware and software resources, human resources and budget (Conboy 2009). Table 2 shows various change aspects and their descriptions.

Aspects of Changes	Description	Key References
Technical Requirement Changes	Changes in technical attributes of the system, such as performance-related attributes, reliability attributes, scalability attributes and availability attributes	(Conboy 2009), (Li et al. 2010)
Business Requirement Changes	Changes in functionalities of the systems that can bring business value to the customer	(Conboy 2009), (Li et al. 2010)
Technological Resource Requirement Changes	Changes (addition or removal) in hardware and software resources which helps IT and Business teams to make system development more effective and efficient.	(Conboy 2009), (Li et al. 2010)
Human Resource Requirement Changes	Changes in human resources with relevant skills which are required to make system development more effective and efficient. i.e. a member left or joined the team	(Boehm and Turner 2005), (Conboy 2009)
Budget and Schedule Changes	Changes in resources (time and budget) for the given requirements. i.e. priority of the requirement changed so need to deliver early	(Conboy 2009), (Vidgen and Wang 2009)

Table 2. Changes and their descriptions

Antecedents of Agility and Team Delivery Capability

Agile software development processes such as communication, collaborative decision making between IT and Business teams, and iterative development process are common across different agile methods. Collaboration with customer or customer representative (e.g., business teams) is needed for various activities during the project such as planning, prioritizing, reviewing, and providing feedback (Hoda et al. 2011). The decisions about various project activities are made after discussions with the IT team and the customer (Drury and McHugh 2011). Agile processes are people centric, so communication is very important (Nerur et al. 2005). It is an important factor for success in agile software development projects because communication problems combined with frequently changing requirements can be unfavorable for the project success (Korkala and Abrahamsson 2007). Iterative development refers to the development of software system in short iterations with continual testing and integration. Delivery strategy has been shown as an important success factor for agile projects (Chow and Cao 2008). Iterative delivery of software and test driven development are key agile processes (Cockburn 2006; Fontana et al. 2014).

Proposed Research Framework

In this study, we conceptualize agility as the outcome of agile processes. Agility is defined as the emergent capability of various agile processes, including communication (Fontana et al. 2014; Korkala and Abrahamsson 2007), collaborative decision making (Batra et al. 2016), and iterative development (Chow and Cao 2008; Hoda et al. 2011). Based on agile literature and a qualitative study on agility facilitators (Batra et al. 2016), we identify these processes as the key determinants of achieving agility and team delivery capability. Our proposed research framework is shown in Figure 1.

Communication and IT-business collaborative decision making are required to plan and deliver customer requirements. They become more critical when there are frequent changes during the project. It is important to collaborate to not only plan and deliver customer requirements, but also to identify and manage the changes that frequently occurring during the development process (e.g., user requirement changes). Therefore, communication and collaborative decision making affect team delivery capability and agility. Feedback of previous iterations from the business teams must be communicated effectively to IT teams so that they can work on the changes suggested in the feedback. Delivering working software through iterative approaches in short iterations reduces wait times for customer feedback. It facilitates anticipating

and responding to requirement changes quickly (Cockburn 2006; Highsmith 2004), which facilitates agility in the project.

Customer satisfaction is achieved when given requirements are delivered and changes are responded to in an effective and efficient manner. Therefore, we propose that team delivery capability and agility affect customer satisfaction. If agility is high, then changes can be implemented effectively and efficiently, which will in turn enhances customer satisfaction. The team delivery capability is an indicator of the amount of resources (time and budget) a team requires to implement the given requirements. Team delivery capability affects project efficiency because if the team has higher delivery capability, then it can implement the given requirements using less resources (time and budget limits). Agility is required to deal with various changes in the project. Team members may require more resources than estimated to deal with changes in the project. Therefore, we propose that agility also affects project efficiency.

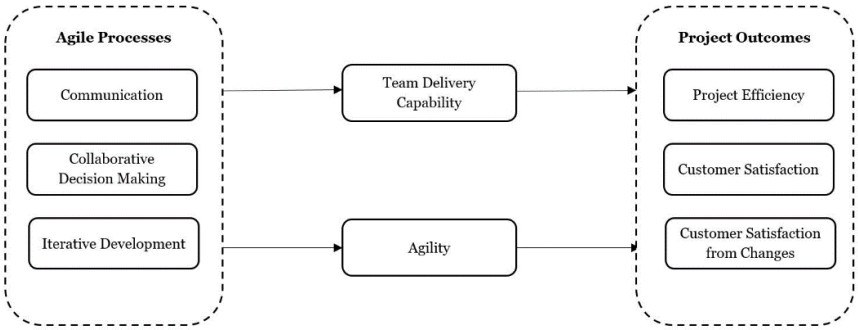

Figure 1. Proposed Research Model

Research Methodology

Wherever appropriate, existing measures were used or adapted for this study. New measures for some constructs were developed based on qualitative analysis of thirteen interviews with agile professionals as well as literature on agile software development. Q-sorting was conducted with five experts for achieving face validity of the measures (Straub et al. 2004). A few items were changed or rephrased after the q-sorting procedures. Next, a pilot test was conducted with twenty-six responses from potential respondents for validating the survey. After the pilot test, a few items were changed, rephrased, or deleted. An online survey method will be used to collect data for this research study from professionals (software developers, business analysts, managers) working on agile projects. The respondents will be approached through online professional communities on social networking sites (e.g., LinkedIn). Survey questions will be measured using a seven point Likert scale. Partial least squares (PLS) technique will be used for survey data analysis.

Expected Contributions and Future Research

This research makes a number of contributions to both IS research and practice. First, this study will identify the agile processes that facilitate agility in a project. IT practitioners need to focus on these processes to enable agility because it is important for dealing with emergent changes in the project. Second, new empirical measures for agility and identified dimensions will contribute to the agile software development literature. Most studies on agile software development are qualitative in nature. It is difficult to understand the multifaceted nature of agility and its relationship with other variables without empirical measures of agility. Finally, the empirical investigation of the relationship between agile processes and agility will help develop a deeper understanding of the theoretical rationale behind agile methodology. The lack of theoretical glue behind agile processes is a key shortcoming (Abrahamsson et al. 2009; Conboy 2009). This study purports to enhance our understanding regarding the theoretical underpinning of agile software development. Future studies may investigate contextual factors that either facilitate or inhibit the adoption and utilization of agile processes. Such contextual factors may include team autonomy, organizational culture, and developer competence.

REFERENCES

Abrahamsson, P., Conboy, K., and Wang, X. 2009. "Lots Done, More to Do: The Current State of Agile Systems Development Research," *European Journal of Information Systems* (18), pp. 281-284.

Balijepally, V., DeHondt, J., Sugumaran, V., and Nerur, S. 2014. "Value Proposition of Agility in Software Development–an Empirical Investigation," in: *20th Americas Conference on Information Systems*. Savannah, Georgia, USA: Association for Information Systems(AIS), pp. 1-14.

Batra, D., Xia, W., and Rathor, S. 2016. "Agility Facilitators for Contemporary Software Development," *Journal of Database Management* (Forthcoming).

Boehm, B., and Turner, R. 2005. "Management Challenges to Implementing Agile Processes in Traditional Development Organizations," *Software, IEEE* (22:5), pp. 30-39.

Chau, T., and Maurer, F. 2004. "Knowledge Sharing in Agile Software Teams," in *Logic Versus Approximation*. Heidelberg: Springer Berlin pp. 173-183.

Chow, T., and Cao, D. B. 2008. "A Survey Study of Critical Success Factors in Agile Software Projects," *Journal of Systems and Software* (81:6), pp. 961-971.

Cockburn, A. 2006. *Agile Software Development: The Cooperative Game*, (2nd ed.). Upper Saddle River, NJ: Pearson Education, Inc.

Conboy, K. 2009. "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research* (20:3), pp. 329-354.

Drury, M., and McHugh, O. 2011. "Factors That Influence the Decision-Making Process in Agile Project Teams Using Scrum Practices," in: *6th International Research Workshop on Information Technology Project Management (IRWITPM)*. Shanghai, China: Association for Information Systems(AIS), pp. 29-40.

Fontana, R. M., Fontana, I. M., da Rosa Garbuio, P. A., Reinehr, S., and Malucelli, A. 2014. "Processes Versus People: How Should Agile Software Development Maturity Be Defined?," *Journal of Systems and Software* (97), pp. 140-155.

Highsmith, J. 2004. *Agile Project Management*. Boston, MA: Addison-Wesley.

Hoda, R., Noble, J., and Marshall, S. 2011. "The Impact of Inadequate Customer Collaboration on Self-Organizing Agile Teams," *Information and Software Technology* (53:5), pp. 521-534.

Korkala, M., and Abrahamsson, P. 2007. "Communication in Distributed Agile Development: A Case Study," in: *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)*. Lubeck, Germany: IEEE Computer Society, pp. 203-210.

Lee, G., and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility," *MIS Quarterly* (34:1), pp. 87-114.

Li, Y., Chang, K. C., Chen, H. G., and Jiang, J. J. 2010. "Software Development Team Flexibility Antecedents," *Journal of Systems and Software* (83:10), pp. 1726-1734.

Lyytinen, K., and Rose, G. M. 2006. "Information System Development Agility as Organizational Learning," *European Journal of Information Systems* (15:2), pp. 183-199.

Maruping, L. M., Venkatesh, V., and Agarwal, R. 2009. "A Control Theory Perspective on Agile Methodology Use and Changing User Requirements," *Information Systems Research* (20:3), pp. 377-399.

Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of Migrating to Agile Methodologies," *Communications of the ACM* (48:5), pp. 72-78.

Sarker, S., and Sarker, S. 2009. "Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context," *Information Systems Research* (20:3), pp. 440-461.

Schwaber, K., and Sutherland, J. 2011. "The Scrum Guide," *Scrum Alliance*, pp. 1-16.

Sheffield, J., and Lemétayer, J. 2013. "Factors Associated with the Software Development Agility of Successful Projects," *International Journal of Project Management* (31:3), pp. 459-472.

Siau, K., Long, Y., and Ling, M. 2010. "Toward a Unified Model of Information Systems Development Success," *Journal of Database Management* (21:1), pp. 80-101.

StandishGroup. 2015. "Chaos Report 2015."

Straub, D., Boudreau, M. C., and Gefen, D. 2004. "Validation Guidelines for IS Positivist Research," *Communications of the Association for Information Systems* (13:24), pp. 380-427.

Vidgen, R., and Wang, X. 2009. "Coevolving Systems and the Organization of Agile Software Development," *Information Systems Research* (20:3), pp. 355-376.

Wendler, R. 2013. "The Structure of Agility from Different Perspectives," in: *Federated Conference on Computer Science and Information Systems (FedCSIS)*. Kraków, Poland: IEEE, pp. 1165–1172.