Contracts in Agile Software Development Projects

Shekhar Rathor
University of Central Oklahoma

Prasad Rudramuniyaiah
University of Central Missouri

ABSTRACT

Agile software development has become very popular in recent years. Agile software development is recommended where requirements are uncertain and project scope evolves with time. As the project scope is not well defined, creating a project contract remains a challenge for most agile practitioners. To address the challenges that agile practitioner face in agile project contracting, there is need for an in-depth understanding of various aspects which makes agile project contracts successful. Based on the extant literature, this study presents a framework for contracts in agile projects.

Keywords: Agile software development, Agile Contracts.

INTRODUCTION

Agile software development practices have become popular in the last decade. According to an industry report, 97% of respondents said that their organizations are using agile practices (VersionOne & Collabnet, 2019). Agile manifesto was released in 2001 to describe values and principles for agile software development (AgileAlliance, 2016). Table 1 in the appendix shows agile values and principles given in the manifesto. While IT practitioners appreciate the benefits of Agile software development, having a contract that supports the agile approach to adopt changes and encourage positive and productive client relationships is a challenge (Eckfeldt *et al.*, 2005). The contracts in software development projects are used to define the working relationship and expectations between the client and the vendor. There are three key goals for any software development contract: to define 1) the purpose of the project, 2) How the project is to be executed 3) what happens if the project goes wrong (Edwards *et al.*, 2012). Many standard traditional contracts focus mainly on the third point while ignoring the importance of the first and second point (Edwards *et al.*, 2012). Both client and vendor in a contract are risk-averse, the client always prefers a fixed price and time contract because it saves them from the uncertainty of budget and time. Such contracts are not suitable for vendors because the scope of the project is not well defined at the beginning of the project. Many software development projects are outsourced. The success of IT outsourcing projects is restricted by the lack of guidance on how to design IT outsourcing contracts, so a comprehensive knowledge of the factors involved in designing contracts is needed to realize the full benefits of outsourcing (Varghese, 2012). Contracting in software projects is a complex process because of the high uncertainty and complexity of software development (Opelt *et al.*, 2013). Agile software development is recommended where requirements are uncertain and project scope evolves with time. The negotiation in agile project contracts is one of the main challenges IT practitioners face (Hoda *et al.*, 2009). To make agile project contracts more effective and less complex, how many details should be included remains a big challenge for most IT practitioners. For example, the agile manifesto recommends frequent communication between client and vendor. There should be some clause/guidelines for communication included in the contract to ensure effective communication or it should not be included because we can't force people to communicate and still call it an agile development environment? More details make a contract complex and complex contract leads to additional cost of contracting (Poppo & Zenger, 2002). Moreover, both the parties involved in outsourcing projects are usually from different cultures and professional backgrounds, so it becomes interesting to study agile project contracts because there may be collaboration issues and a lack of trust.

Many studies have been done on IT outsourcing contracts but most of them focus on contracts in traditional projects in which scope, time, and cost are usually defined at the beginning of the project. We believe that the research related to contracts in agile software projects is limited, and prior literature has recognized the gap for studies related to such contracts (Edwards *et al.*, 2012; Hoda *et al.*, 2009). To address the challenges that practitioner face in agile project contracting, there is need for an in-depth understanding of various aspects which makes agile project contracts successful. Based on the extant literature, this study presents a framework for contracts in agile projects.

AGILE CONTRACT FRAMEWORK

Software contracting is a multi-faceted issue that involves legal, economic, managerial, and technological considerations (Whang, 1992). The structural and legal aspects of agile project contracts and traditional project contracts are the same, the differences lie in the approach to and understanding of the operational process and delivery and how this is captured in or intersects with contracts (Larman & Vodde, 2010). The two common forms of contracts are fixed-price (FP) and time-and-materials (TM) contracts (Chen & Bharadwaj, 2009; Gopal *et al.*, 2003). A contract represents promises or obligations to perform particular actions in the future (Macneil, 1977). The specification of promises, obligations, and processes for dispute resolution is more when project contracts are more complex(Poppo & Zenger, 2002). More complex contract leads to additional cost of contracting (Poppo & Zenger, 2002). Agile manifesto emphasizes focusing more on customer collaboration than on contract negotiation. Relational governance, which is based on trust & agreed-upon processes minimizes transaction costs and facilitates adaptive responses (Poppo & Zenger, 2002). For outsourcing projects, legal experts suggest that a contract should include at least a specified service level agreement, penalty clauses, specific arrangements for changing circumstances, and early termination clauses (Fitzgerald & Willcocks, 1994). The contracts can promote the expectation that the other party will behave cooperatively because of contractual safeguards but in uncertain business situations, it is not able to maintain cooperation and relationship(Poppo & Zenger, 2002). Usually, most agile projects are based on uncertain business requirements so agile project contracts are designed to embrace future uncertainties.

The future uncertainties lead to incomplete contracts which can't embrace all possible contingencies(Hart, 1988). The contract can be used as a governance tool for relationships and outsourcing success can be enhanced by having negotiation and change request clauses (Gellings, 2007). The research studies have shown conflicting views on the relationship between formal contracts and relational governance. Some research studies view relational governance and formal contracts as a substitute to each other (Dyer & Singh, 1998). The trust between parties reduces fear of opportunistic behavior, reduces transaction costs, lowers the need for monitoring, and facilitates contractual adaptation (Gulati, 1995). Some studies state that formal contracts and relationship governance complement each other (Poppo & Zenger, 2002). The findings related to the traditional contracts may not apply to agile contracts. To understand the contracting in agile projects, we present a framework for agile contracts and discuss the key criteria for contract choice, dimensions of agile contacts, and the outcome of the project in the following sections. Figure 1 in the appendix shows the agile contract framework.

Contract Choice

In this framework, we argue that the choice of the contract type is critical for the success of agile projects. There are many prevalent contract types in the IT industry that use a hybrid approach for agile projects. A blog by an agile coach talks about ten contract types that can work for agile projects (Arbogast *et al.*, 2012). The type of contract framework used in a software development project can have a significant impact on the project price, business value from the project, and on the completion probability of the project (Richmond & Seidmann, 1993). The study by Fitzgerald and Willcocks identified six types of contracts for IT outsourcing (Fitzgerald & Willcocks, 1994). Franklin (2008) talks about the evolution of agile contracts in a company and states that agile development practices can be successfully done under contracts with fixed

constraints, schedule, and cost. The characteristics of the software project such as requirement uncertainty, project size (team hours), project importance, and presence of other vendors are critical for the choice of the contract (Fink & Lichtenstein, 2014; Gopal *et al.*, 2003). Therefore, it is critical for agile practitioners to choose a contract type that is appropriate based on the project characteristics.

CONTRACT DIMENSIONS

In agile projects, the scope of the project evolves with each delivery cycle based on customer feedback (Nerur *et al.*, 2005). Traditional project contracts mainly focus on cost, time, and scope, while agile project contracts focus more on customer value and sharing risks. Agile values recommend prioritizing customer collaboration over contract negotiation. In this framework, we present contract dimensions that are critical for project success. These dimensions are flexibility, collaboration, and transparency.

Flexibility

The clients prefer a fixed bid contract with fixed time, cost, and scope of the project because it provides them with a perceived sense of predictability and control over the project schedule, cost, and deliverables (Hoda *et al.*, 2009). One of the agile principles recommends welcoming requirement changes even late in the project which creates uncertainty as project scope can change in a future iteration. Project uncertainty has a negative effect on the project's success (Jiang *et al.*, 2002; Jun *et al.*, 2011). The changes that arise during the project are one of the major causes for low project performance in software development projects. (Jun *et al.*, 2011; Wang *et al.*, 2016). The agile software development methodology is recommended for projects where requirements are uncertain, and the scope of the project evolves with time. Due to this reason scope, time and budget are not very clearly defined at the time of making a contract for agile projects. Since the scope of projects is not clearly defined, it is not possible to create a contract that discusses all possible scenarios of the project. A contract with a time and money model which allows flexibility to develop software incrementally can be an ideal approach for an agile contract (Franklin, 2008). IT project contracts should be flexible for changes (Qi & Chau, 2015). Keeping in view the unclear future needs, agile project contracts should be designed to embrace changes, which helps the client and vendor in achieving the desired project goals. Therefore, we argue that flexibility is an important dimension of agile contracts.

Collaboration:

Agile projects are not successful just because of contracts but also because of the relationship between client and vendor (Opelt *et al.*, 2013). It is difficult to cover all possible scenarios in the contract when the project scope can change so a collaborative relationship between client and vendor is critical for the project's success. In Agile projects, the scope of the project is not completely defined at the beginning of the project. Because of changing scope fixed bid projects are not preferred by the vendor because it is risky for them to commit to a project with a fixed amount whose scope is not clear in the beginning. Strategies like changing the client's mindset and providing different options of working are important for the vendor to handle problems related to negotiation in agile project contracts with clients (Hoda *et al.*, 2009).

Traditional project contracts focus more on indemnity because of the long delay in getting working software, poor and late feedback process whereas, agile contracts focus more on the business value of the product under development (Opelt *et al.*, 2013). It is because of an iterative approach to deliver working software and early feedback process (Opelt *et al.*, 2013). The client can terminate the project in a very short time (two to four weeks) if something goes wrong, which is not possible in the traditional non-agile project (Larman & Vodde, 2010). The collaborative contracts which include sharing risks and benefits between vendor and client can successfully work in agile projects (Thorup & Jensen, 2009). Agile contracts can be used to define a collaborative framework to achieve goals of agility and enhance project success (Opelt *et al.*, 2013). Collaboration between client and vendor is one the key aspect which makes an agile project successful (Batra *et al.*, 2017; Cockburn, 2006). Agile principles recommend that IT and Business people should work together and should communicate frequently during the project. The guidelines and clauses are given in the agile project contract regarding having a shared vision, communication between client and vendor, collective responsibilities and knowledge sharing can help in achieving collaboration between client and vendor. Therefore, collaboration is an important dimension of agile contracts.

Transparency

The two main purposes of contracts are, a) to protect each party from opportunistic behavior on the part of the other party, b) to set up appropriate incentives for companies to work together in a synergistic matter (Poppendieck & Poppendieck, 2005). These purposes are mainly related to trust between client and vendor, so when there is not sufficient trust between the client and the vendor, one would likely try to prevent one another from opportunistic behavior (Ganes & Nævdal, 2008). The trust between client and vendor is very critical for the success of a software development project (Sabherwal, 1999). The contract between client and vendor acts as a base for relationship building (Goo & Nam, 2007; McFarlan & Nolan, 1995; Qi & Chau, 2015). In an agile project, there is frequent communication between client and vendor, so the client is frequently updated about the progress and developmental issues. This transparency in the working culture leads to trust between client and vendor. Information sharing and communication quality are necessary to developing trust between client and vendor (Mao *et al.*, 2008). In the outsourcing scenario, usually, clients and vendors are from a different culture without any prior working experience with each other. Agile contracts can be designed to set up mechanisms that help in building transparency between the client and the vendor. Most agile methods have well-defined working rules and processes (Edwards *et al.*, 2012). For example, Scrum clearly defines the role of product owner, scrum master, etc., and activities such as sprint planning and retrospective meetings. Due to clear definitions of roles and activities, most agile methods promote working visibility between both parties. Agile contracts focus on promoting visibility and rather the legal remedies (Edwards *et al.*, 2012). Therefore, transparency is an important dimension of agile contracts.

Project Success

The contract choice significantly determines project outcomes (Gopal *et al.*, 2003). Stakeholders must choose contracts that facilitate project success. The meaning of project success can be different depending upon the stakeholder's perspective (Thomas & Fernández,

2008). Project success can be defined in terms of system quality (Cao *et al.*, 2009; Sheffield & Lemetayer, 2013), business value of the system (Sheffield & Lemetayer, 2013; Siau *et al.*, 2010), system functionality (Lee & Xia, 2010; Siau *et al.*, 2010), on-time completion (Cao *et al.*, 2009; Lee & Xia, 2010), within-budget completion (Cao *et al.*, 2009; Lee & Xia, 2010), and customer satisfaction (Sheffield & Lemetayer, 2013; Siau *et al.*, 2010).

In this framework, we present three dimensions that are critical for project success. The contract dimensions have an impact on project success (Qi & Chau, 2015). We present project success in terms of client and vendor satisfaction. If the client and vendor achieved their intended goals, then we can say that the project is successful. If the contract is flexible to allow changes especially requirement changes in the project, then it will enhance the satisfaction of the stakeholders. Usually, IT teams represent the vendor side and Business teams represent the client side. The implementation of requirement changes results in better business value output which enhances client satisfaction. When IT team implements changes that are critical for the business value, it enhances their satisfaction also (Tripp *et al.*, 2016). The contracts that allow change management positively influence project success and trust between client and vendor (Goo & Nam, 2007). The collaborative work culture and transparency in working helps in building a trust relationship between client and vendor.

CONTRIBUTIONS AND FUTURE RESEARCH

This study has many contributions to agile literature and for agile practitioners. Firstly, this study contributes to the literature by providing important insights into the contracts in agile projects. There is a scarcity of studies related to agile contracts. This study is an important step towards understanding agile project contracts. Secondly, this study can help clients to change their perception of agile project contracts by explaining that agile project contracts should not be treated as traditional projects. Unlike traditional projects, the client needs to collaborate with the vendor proactively and a contract can help in achieving this by defining a collaborative framework and rules of engagement. Changing the customer's mindset is one of the major challenges faced by IT practitioners in agile project contracts (Hoda *et al.*, 2009). Most clients expect a detailed contract that will protect them from vendor's opportunistic behavior. Such contracts are not possible when a client is asking for a product based on incomplete project scope and uncertain business requirements. Finally, this study presents a framework for the success of an agile project and contract. It presents dimensions that are critical for the project's success. Agile practitioners can use this framework to develop contracts for agile projects because, unlike traditional project contracts, agile project contracts can't have everything written in a contract as the project scope is not well defined at the time of the contract. Future studies can empirically test the validity of the presented framework. Quantitative studies can be conducted by collecting survey data to develop additional insights about agile contracts.

REFERENCES

AgileAlliance. (2016). What is Agile? Retrieved from https://www.agilealliance.org/agile101/what-is-agile/

Arbogast, T., Larman, C., & Vodde, B. (2012). Agile contract primer. Retrieved from http://www.agilecontracts.org/agile_contracts_primer.pdf

Batra, D., Xia, W., & Zhang, M. (2017). Collaboration in Agile Software Development: Concept and Dimensions. *Communications of the Association for Information Systems, 41*(1), 20

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems, 18*(4), 332-343

Chen, Y. Y., & Bharadwaj, A. (2009). An Empirical Analysis of Contract Structures in IT Outsourcing. *Information Systems Research, 20*(4), 484-506

Cockburn, A. (2006). *Agile software development: the cooperative game* (2nd ed.). Upper Saddle River, NJ: Pearson Education, Inc.

Dyer, J. H., & Singh, H. (1998). The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review, 23*(4), 660-679

Eckfeldt, B., Madden, R., & Horowitz, J. (2005). Selling agile: target-cost contracts. *Proceedings of the Agile Development Conference (ADC'05).*

Edwards, I., Bickerstaff, R., & Duisberg, A. (2012). Contracting for Agile software development projects. Retrieved from https://www.twobirds.com

Fink, L., & Lichtenstein, Y. (2014). Why project size matters for contract choice in software development outsourcing. *The DATABASE for Advances in Information Systems, 45*(3), 54-71

Fitzgerald, G., & Willcocks, L. (1994). Contracts and Partnerships in the Outsourcing of IT. *Proceedings of the International Conference on Information Systems* Vancouver, British Columbia, Canada.

Franklin, T. (2008). Adventures in agile contracting: Evolving from time and materials to fixed price, fixed scope contracts. *Proceedings of the Agile 2008 Conference.*

Ganes, A., & Nævdal, S. (2008). *Software Contracting and Agile Development in the Norwegian ICT Industry: A Qualitative Survey.* Institutt for datateknikk og informasjonsvitenskap.

Gellings, C. (2007). Outsourcing relationships: the contract as IT governance tool. *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* Hawaii, USA.

Goo, J., & Nam, K. (2007). Contract as a source of trust--commitment in successful IT outsourcing relationship: an empirical study. *Proceedings of the Annual Hawaii International Conference on System Sciences*, Hawaii, USA.

Gopal, A., Sivaramakrishnan, K., Krishnan, M. S., & Mukhopadhyay, T. (2003). Contracts in offshore software development: An empirical analysis. *Management Science, 49*(12), 1671-1683

Gulati, R. (1995). Does familiarity breed trust? The implications of repeated ties for contractual choice in alliances. *Academy of Management Journal, 38*(1), 85-112

Hart, O. D. (1988). Incomplete Contracts and the Theory of the Firm. *Journal of Law, Ecomonics and Organization, 4*, 119

Hoda, R., Noble, J., & Marshall, S. (2009). Negotiating contracts for agile projects: A practical perspective. *Proceedings of the Agile Processes in Software Engineering and Extreme Programming*, Pula, Italy.

Jiang, J. J., Klein, G., & Ellis, T. S. (2002). A measure of software development risk. *Project Management Journal, 33*(3), 30-41

Jun, L., Qiuzhen, W., & Qingguo, M. (2011). The effects of project uncertainty and risk management on IS development project performance: A vendor perspective. *International Journal of Project Management, 29*(7), 923-933

Larman, C., & Vodde, B. (2010). *Practices for scaling lean & agile development: large, multisite, and offshore product development with large-scale Scrum*: Pearson Education.

Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly, 34*(1), 87-114

Macneil, I. R. (1977). Contracts: Adjustment of long-term economic relations under classical, neoclassical, and relational contract law. *Nw. UL Rev., 72*, 854

Mao, J.-Y., Lee, J.-N., & Deng, C.-P. (2008). Vendors' perspectives on trust and control in offshore information systems outsourcing. *Information & Management, 45*(7), 482-492

McFarlan, F. W., & Nolan, R. L. (1995). How to manage an IT outsourcing alliance. *MIT Sloan Management Review, 36*(2), 9

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 48*(5), 72-78

Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile contracts: creating and managing successful projects with Scrum*: John Wiley & Sons.

Poppendieck, M., & Poppendieck, T. (2005). Agile contracts *Extreme Programming and Agile Processes in Software Engineering* (pp. 302-302): Springer.

Poppo, L., & Zenger, T. (2002). Do formal contracts and relational governance function as substitutes or complements? *Strategic Management Journal, 23*(8), 707-725

Qi, C., & Chau, P. Y. (2015). Relationship or contract? Exploring the key factor leading to IT outsourcing success in China. *Information Technology & People*

Richmond, W. B., & Seidmann, A. (1993). Software development outsourcing contract: Structure and business value. *Journal of Management Information Systems 10*(1), 57-72

Sabherwal, R. (1999). The role of trust in outsourced IS development projects. *Communications of the ACM, 42*(2), 80-86

Sheffield, J., & Lemetayer, J. (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management, 31*(3), 459-472

Siau, K., Long, Y., & Ling, M. (2010). Toward a unified model of information systems development success. *Journal of Database Management (JDM), 21*(1), 80-101

Thomas, G., & Fernández, W. (2008). Success in IT projects: A matter of definition? *International Journal of Project Management, 26*(7), 733-742

Thorup, L., & Jensen, B. (2009). Collaborative agile contracts. *Proceedings of the Agile Conference*, Chicago, USA

Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems, 17*(4), 267

Varghese, S. P. (2012). Agility in Fixed-Price Projects. Retrieved from https://www.projectmanagement.com/articles/283717/Agility-in-Fixed-Price-Projects

VersionOne, & Collabnet. (2019). *13th Annual State of Agile Report*

Wang, S.-Y., Hsu, J. S.-C., Li, Y., & Cheng, K.-T. (2016). Team Quotients, Resilience, and Performance of Software Development Projects.

Whang, S. (1992). Contracting for software development. *Management Science, 38*(3), 307-324

APPENDIX

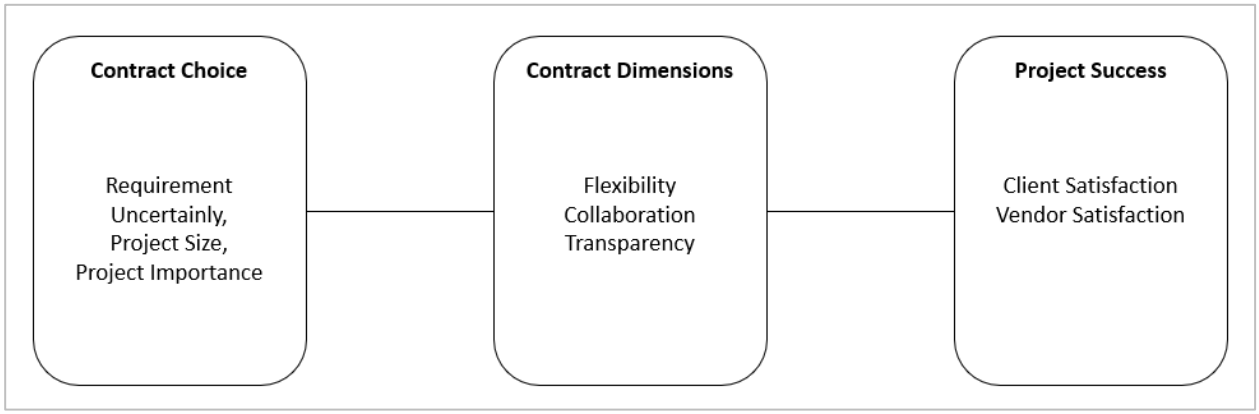

Figure 1. Agile Contract Framework

Table 1. Agile Values and Principles

Agile Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to, and within, a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.