Chapter 33
Traditional or Agile Contracting for Software Development:
Decisions, Decisions

Dinah Payne
University of New Orleans, USA

ABSTRACT

As the use of software is present in so many activities today, it is important for business in particular to be aware of challenges that may seem different today than before the prevalence of software in our lives. Agile project management is one example: this more recent and nimble approach to software development presents its own challenges. Fortunately, the guiding legal principles related to traditional contract formation and execution are based in principles of fairness and equity, making the customization of legal principles to Agile contracting a reasonable endeavor. This chapter presents basic contract law and such law as it more specifically relates to contracts dealing with Agile software development.

INTRODUCTION: WHY DOES THE CONTRACT MATTER?

According to the Legal Executive Institute, United States (U. S.) companies spend approximately 40% of their revenues on legal services (2017). This is an astonishing number and considerably more than companies in other venues. Miller (2015) reports that CEOs and CFOs spend a great deal of their time on legal matters, from educating outside counsel about the business and the issues it confronts, to seeking to find information requested as a result of legal action, and to preparing for and complying with requests for discovery (depositions, interrogators and requests for documents). These managers must help prepare expert witnesses acting on behalf of the firm and attend hearings or go to trials. Litigation, then, is very expensive and time consuming. "Litigation …will reach deep into the business and the company needs to prepare for and accept that a number of different and valuable people will be taken away from big parts of their day jobs to assist with the (legal) effort (Miller, 2015)." One way to reduce the amount of time and energy wasted on legal disputes is to obviate their occurrence and one way to do this is to

fashion carefully researched and crafted contracts. This is no less true for Agile contracting than it is for any other business contracting. While Agile contracting may involve a higher degree of collaboration than tradition contracts, the need for trust and the need for as much specificity as possible, even in light of the existence of known uncertainties, is critically important. Software developers and owners must accept that there will be disputes arising out of their contracts: the best way to reduce the negative effects of these disputes is to anticipate as much as possible of what might be points of contention.

There is not yet a great deal of litigation involving project management using the Agile approach, so the development of this material is based on general contract law, which we will see is based on legal principle that works for traditional or Agile contracting. However, in 2019, there was an instance of a dispute that resulted in a split decision by a Texas court (Raysman and Brown, 2019). In *Polar Pro Filters Inc v. Frogslayer, LLC* (2019), the dispute arose from a software development agreement that ran significantly over the original cost estimate and over the revised cost estimate. Further, fraud was claimed as a result of the non-delivery of the software in a viable form. The developer did not receive what he considered to be his full payment and the owner did not receive a viable product. The claims in the case were all based on traditional contract law even though the subject matter was a contract for software development wherein iterations were to be used. The point is that, although traditional contract and Agile contract may focus on, for example, different kinds of payment schedules or different visions of what should be delivered at what times, what installments or iterations are due/owing, the law used to generate a fair and equitable resolution to the dispute is well-established contract law. As Agile contracts become more frequently used and as more litigation develops, it is certainly possible that specific rules will be developed for Agile contract attributes that warrant different treatment. At this time, however, as evidenced by the *Polar Pro* case, traditional notions of contract essentials like fraud and breach of contract have served Agile contracting as well as traditional contracting.

In every successful business relationship, the parties must have agreed as to the purpose of the relationship. Each party must have a reasonably definite concept of what his obligations are under any contract he enters into: this is axiomatic in business and in business law. In traditional contracts, there is certainly enough litigation to make the retention of an attorney a good idea to give the parties a path to redress alleged transgressions, as well as to assure pro-active work to prevent any legal conflict via a well-written contract.

In Agile contracts, the pro-active nature of the work is even more important, as the relationship between the owner of the project and the software developer is far more collaborative than unilaterally providing goods between the buyer and seller. "The unprecedented rate of change in business and technology has made it increasingly difficult for software teams to determine user requirements and respond to their changes…Agile development approaches differ from the traditional, plan-driven, structured approaches as the former put more emphasis on lean processes and dynamic adaptation than on detailed front-end plans and heavy documentation (Lee and Xia, 2010: 88)." This description of Agile project management itself is a description of the root of the difference in contract law between traditional contracts and Agile contracts: as the word denotes, agile means to move quickly and easily (Dictionary.com, n.d.).

The Agile form of project management, designed to be rapid and easily coordinated, is represented by a framework for organizing and managing work in iterative stages. This differentiates Agile contracts from others in a number of ways, not the least of which is that, by their nature, the structure and stability generally found in contracts alerting the contracting parties as to their obligations is not possible: since the work is done on an "as we go" basis, iteratively, concepts of performance, for example, are different between traditional contract law and the contract law associated with Agile. We present here basic

Related Content

Framework-Based Debugging for Embedded Systems
Gokhan Tanyeri, Trish Messiterand Paul Beckett (2014). *Handbook of Research on Embedded Systems Design (pp. 424-454).*
www.irma-international.org/chapter/framework-based-debugging-for-embedded-systems/116121

Towards Risk Based Effort Estimation: A Framework to Identify, Analyze, and Classify Risk for Early Identification at Requirement Engineering Phase
Cerene Mariam Abraham, M. Sudheep Elayidomand T. Santhanakrishnan (2018). *International Journal of Information System Modeling and Design (pp. 67-84).*
www.irma-international.org/article/towards-risk-based-effort-estimation/220458

WSN-Driven Posture Recognition and Correction Towards Basketball Exercise
Xiangyang Cai (2022). *International Journal of Information System Modeling and Design (pp. 1-14).*
www.irma-international.org/article/wsn-driven-posture-recognition-and-correction-towards-basketball-exercise/300777

Principles of Modeling in Information Communication Systems and Networks
Oleg I. Sheluhinand Aderemi A. Atayero (2013). *Integrated Models for Information Communication Systems and Networks: Design and Development (pp. 1-15).*
www.irma-international.org/chapter/principles-of-modeling-in-information-communication-systems-and-networks/79656

Estimation of Target Defect Prediction Coverage in Heterogeneous Cross Software Projects
Rohit Vashishtand Syed Afzal Murtaza Rizvi (2021). *International Journal of Information System Modeling and Design (pp. 73-93).*
www.irma-international.org/article/estimation-of-target-defect-prediction-coverage-in-heterogeneous-cross-software-projects/273227