Chapter 12

Fixed Priced Projects in Agile:
Fixed Projects in Agile Software Development Environments

Anuradha Chaminda Gajanayaka
Exilesoft (Pvt) Limited, Sri Lanka

ABSTRACT

Agile software development has established as a reliable alternative to waterfall software development model. Unfortunately the use of agile software development has been limited to time based contracts and not for time limited contracts. The main reason for this limitation is the “Agile manifesto” itself. The forth value of the manifesto states that agile believers find more value in “Responding to change over following a plan”. This is one of the main reasons why agile software development methods are not preferred for a fixed priced contract or time limited contract. The following case study provides an example on how the agile software development can be used for fixed priced software development contracts even when operating in offshore context. The agile software development concepts were used throughout to plan, execute, monitor, report, etc. for the project documented in this case study.

INTRODUCTION

Fixed Projects in Agile Software Development Environments

A fixed priced software development contract requires fixing all project parameters before the start of the project. There are three main project parameters referred as triple constraints (Zhang, 2008). They are (a) scope, (b) cost, and (c) schedule. There is a considerable upfront effort to fix the triple constraints and the outcome of this effort is the “project plan”.

The main rational behind fixing the parameters upfront is to transfer the risk of scope/cost/schedule overruns to supplier. This approach has been criticized by many leading agile practitioners. (S. W. Ambler, 2003) argues that “Risk should be borne by the party best able to manage it” (p.158). There are many disagreements between how the agile thinkers want to run a project and what a fixed priced contracts demands.

DOI: 10.4018/978-1-4666-9858-1.ch012
The first major conflict can be found in the agile manifesto itself. The fourth value of the manifesto states agile believers fine more value in “Responding to change over following a plan”. Further, the second principle of the manifesto states that agile believers welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. This is in direct conflict of fixing triple constraints before the start of the project.

The second important conflict is that agile believers do not encourage big upfront planning effort to fix the project parameters. Agile thinkers believe this upfront effort is a waste. Mike Cohn (2011) explains how Scrum is not fighting the so called cone of uncertainty (p. 3). Agile followers believe in small planning step at the start and then iteratively improve the plan when the project moves forward. (Cohn, 2006) mentions: what is important is the “planning not the plan” (p. 9)

The result of these conflicts is creation of the common notion that agile software development is not suitable for fixed priced contracts. Fixed priced contracts are being managed using waterfall software development model(Schwaber, 2004) argues that scrum is not a silver bullet (Hoda, Noble, & Marshall, 2009). The downside of this has been fixed priced contracts do not get the benefits which agile software development offers.

It is not always possible for an agile software development organization to turn down all the fixed priced projects. Situations such as government contracts, tenders, etc. require fixing the triple constraints before the start of the project.

But certain sections of agile community are being in favor of using agile software development in fixed contracts. In one example, (Fowler, 2003) argues that is it is possible to use agile methods in fixed price contracts.

“There are two major challenges when applying agile methodologies in to a fixed price contract. One is how to fix the three parameters at the beginning and the second challenge is how to manage the fixed project while not violating agile values and principles. The second reason is how to manage the project once the parameters are fixed.” (Fowler, 2003).

Any of these discussions has matured to result in a meaningful framework on how to apply agile software development in fixed priced contracts. This case study demonstrates how the project team managed to use agile software development concepts in a fixed priced contract. Team had to face many additional complexities than fixing triple constraints.

- The scope was to re-write a legacy system which was in production for number of years. The data migration & new system transition have to happen without affecting a single user
- Team had only seven working days to provide the project proposal
- Team and the client were geographically separated by 8000 kilometers and they didn’t know each other before
- The time difference between the client and the supplier was 4.5 hours

The team involved in the project had 3 software developers, one project manager and one consultant architect. The consultant architect was able to locate himself at the client’s site during the 7 days period where the proposal was made (Agile Methodology in Fixed Price projects. Global Advanced Research Journal of Engineering, 2013).

This was an offshore software development project using agile software development practices to manage & successfully deliver a fixed price contract.
Related Content

Swap Token: Rethink the Application of the LRU Principle on Paging to Remove System Thrashing
[www.igi-global.com/chapter/swap-token-rethink-application-lru/37945?camid=4v1a](www.igi-global.com/chapter/swap-token-rethink-application-lru/37945?camid=4v1a)

Object-Aware Business Processes: Fundamental Requirements and their Support in Existing Approaches
[www.igi-global.com/article/object-aware-business-processes/53204?camid=4v1a](www.igi-global.com/article/object-aware-business-processes/53204?camid=4v1a)

Round-trip Engineering UML Class Models and Java Models: A Real-world Use Case for Bidirectional Transformations with QVT-R
[www.igi-global.com/article/round-trip-engineering-uml-class-models-and-java-models/170520?camid=4v1a](www.igi-global.com/article/round-trip-engineering-uml-class-models-and-java-models/170520?camid=4v1a)

Optimized and Distributed Variant Logic for Model-Driven Applications
[www.igi-global.com/chapter/optimized-and-distributed-variant-logic-for-model-driven-applications/117936?camid=4v1a](www.igi-global.com/chapter/optimized-and-distributed-variant-logic-for-model-driven-applications/117936?camid=4v1a)