Fixed Price Distributed Agile Projects in Outsourced Product Testing – An Experience Report

Raja Bavani
MindTree Ltd., Pune-411057, India
raja_bavani@mindtree.com

Abstract: Agile Software Development and the breed of Agile Methodologies (XP, SCRUM, DSDM, etc.) have gained popularity since 2001. Primarily founded as methodologies for software projects executed at a single location, Agile Methodologies have started showing promising results in multi-site projects too with many adopters and practitioners across the globe. Agile Software Development focuses on early delivery of working software to measure the progress of projects and hence to mitigate risks. It creates an environment that responds to changes by means of being flexible and nimble. It discourages creation of extensive documents that do not add any value to the customer. Distributed Agile Software Development and Testing is nothing but applying Agile Principles and Practices to software projects executed by teams located across geographies. In this context, service providers prefer Time & Material based pricing over Fixed Price model in order to avoid the risk of cost overruns induced by project changes. On the other hand, in order to reduce financial risks and to implement a fixed-scope on a mutually agreeable timeline or fixed-schedule, customers prefer to execute projects on fixed-cost basis. This paper is based on our experience in delivering a Fixed Price Distributed Agile project to one of our strategic customers. This paper presents the challenges in executing Fixed Price contracts, and the approach and real time course corrections we made to deliver this project. Besides, this paper discusses our Findings along with the Best Practices and Lessons Learned.

1. INTRODUCTION

In Outsourced Product Development & Testing arena Agile [1] has been the mantra of success during this decade. This trend is going to continue as the adoption of agile practices continues to increase with seemingly better rate of adoption. In order to exploit the availability of talent mix and associated cost-effectiveness in various geographies, distributed agile has started gaining momentum [4]. When it comes to executing multi-site projects, agile practitioners do face several challenges due to factors such as distribution of teams that results in limitations in face-to-face communication, and cultural mix of teams that impedes team bonding. The goal of agile teams is to deliver working software at frequent intervals, maintain sustainable pace and respond to change. With all these factors, executing agile projects in Fixed Price model poses challenges in terms of managing changes related to project scope, technology as well as quality expectations [8]. On the other hand, projects are sanctioned based on pre-approved budgets and customers prefer to get projects done in fixed-cost. This paper, though it does not attempt to address all the nuances of executing Fixed Price Distributed Agile projects, intends to present our experience in terms of Best Practices, Lessons Learned as well as Findings gathered through executing a Performance Testing project in Fixed Price model.

2. ABOUT THE PROJECT

We executed this project for one of the leaders in Software Product Engineering space. The objective of this project was to a) conduct Performance Testing of one of the key products of our customer under multiple hardware configurations, b) offer recommendations on optimal database configurations, and c) submit a comprehensive performance test report. The core functionality of this product involved enterprise wide security management across several products and applications.

We did not have any internal expertise in comprehending this requirement at full length and breadth. However, our customer offered to provide us adequate support to help us understand the product and wanted us to accomplish this objective in Fixed Price model in order to provide the right solution to end users. The challenge in front of us
was to understand the product as well as project requirements and submit a Fixed Price proposal.

3. OUR APPROACH

Our Solution Architect and Performance Engineering Lead worked with our customer’s Subject Matter Experts over several meetings and conference calls to understand the product functionality and project requirements. This helped us derive the initial scope and identify
- the list of hardware configurations,
- product resources,
- the list of use cases,
- performance parameters, and
- tools.

The scope of this project included peak and steady state performance measurement and analysis of parameters such as CPU usage, memory consumption, response time, database connections, number of connections processed per unit time and number of interceptions processed per unit time for a pre-determined set of transactions. Also, during our discussions our customer wanted us to develop and deliver a Performance Testing framework that will include mechanisms for data creation, load generation, data gathering and analysis, and reporting.

Executing such a project in Fixed Price model required a strong strategy and hence we decide to propose a phased approach so that each phase can be executed in a Fixed Price model as shown in Table I.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Duration &amp; Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase I - Strategy and Planning</td>
<td>2 weeks, Customer Site</td>
<td>Product Demo, Hands-on Learning, Discussions, Strategy Formulation, and High Level Planning</td>
</tr>
<tr>
<td>Phase II - Execution (Framework Development, Test Design, Test Execution, Analysis and Reporting)</td>
<td>22 weeks, Distributed Locations</td>
<td>11 iterations of 2 weeks each with daily stand-ups and iteration reviews &amp; retrospectives</td>
</tr>
<tr>
<td>Phase III - Training &amp; Closure</td>
<td>1 week, Customer Site</td>
<td>Training on Performance Testing Framework and Project Closure</td>
</tr>
</tbody>
</table>

We formed a steering committee that comprised of representatives from our organization as well as our customer’s organization. This enabled us to mutually agree on the dependency factors such as a) availability and stability of the product as well as the test environment, b) availability of test data, c) remote access to test environment, d) involvement of Subject Matter Experts, and e) support from IT staff. By studying the high level characteristics of this project our Solution Architect came up with a phase wise Fixed Price proposal that can be refined with mutual consent at the end of each phase. In order to arrive at a Fixed Price proposal we used experience-based estimates using Wideband Delphi Technique for different phases. After multiple rounds of discussions with our customer we mutually agreed to submit a combined Fixed Price quote for Phase II and Phase III at the end of the first phase.

4. TEAM COMPOSITION

The team composed for this project consisted of a Technical Project Manager, 2 Performance Test Automation Engineers, 1 Performance Test Engineer and 1 Database Expert. Besides, the team included 2 Test Engineers who were Subject Matter Experts from the customer organization. This project required engineers with niche skills and they were not available at a single location. Hence, we formed a distributed team as depicted in Fig. 1.

5. PHASE I – STRATEGY AND PLANNING

The objective of this phase was to come up with a Performance Test Strategy and a high level plan. All team members except the Database Expert had face-to-face discussions during this phase in order to formulate a viable strategy. The Database Expert collaborated with our team through conference calls and emails during this phase. By the end of this phase we could
- refine the project scope,
- validate 95% of the assumptions,
- access risks and identify issues,
- derive a high level plan of subsequent phases, and
- identify infrastructure requirements.

We started this phase and ended it on schedule. We met all objectives of this phase and delivered the Performance Test Strategy document. Successful distributed agile projects happen because of collaborative teams that drive to define a methodology for themselves [4]. The definition of such a methodology happens by means of open communication and minor adjustments that make things work as expected [5]. More importantly, a methodology that works for one distributed ecosystem may not work for another distributed ecosystem. This is because, for any methodology, while the basic tenets remain intact, the implementation details vary across ecosystems [2]. With several years of experience in executing projects with distributed teams we collaborated with our customer and put together a set of lightweight processes to execute this project. The following sections in this paper discuss some of these processes we implemented during subsequent phases.

6. Phase II – Execution

This phase included activities such as Framework Development, Test Design, Test Execution, Analysis and Reporting. We collaborated with our customer in Iteration Planning and implemented Daily Standup Meetings and Iteration Reviews and Retrospectives.

A. Daily Standup Meetings

In case of agile projects, collocated teams and face-to-face communication enable high level of collaboration. Collocated teams can have face-to-face meetings and discussions to iron out the base architecture, design guidelines and understand critical milestones. On the other hand, in case of distributed agile projects setting up the base camp [9] is very essential in order to put the right foot forward. We did this during Phase I by means of involving all key team members in joint discussions to arrive at a viable strategy and high level plan. In order to manage the iterative execution of subsequent phases, we suggested daily standup meetings that are very similar to 15-minute daily meetings suggested by Scrum [1]. Daily standup meetings helped us resolve issues in a timely manner. Also, team members were able to understand the work progress and task allocation of everyone in the team. This increased collaboration and mutual support among distributed teams. Our customer was very supportive of agile processes [1] and encouraged daily standup meetings. As the teams were distributed, we conducted daily standup meetings over conference calls.

Daily Standup Meetings provided us several benefits. On project execution front these meetings provided us an opportunity to have adequate clarity on task execution, impediments, as well as stability of technical infrastructure, tools and key engineering processes [3]. On people front these meetings provided us an opportunity for rapport building and hence enabled efficiency in issue resolution and conflict management throughout the project.

B. Iteration Planning

The decision to implement a phased approach motivated us to understand the project scope in terms of use cases to be tested on a pre-defined set of environments. We suggested Iteration Planning meetings at the start of iterations in order to prioritize and select user stories and related tasks that can be accommodated in a time box of 2 weeks. During the second iteration we came across severe challenges as the test environment was not getting ready as expected. There were technical issues as well as operational issues. Overall, Iteration Planning meetings proved to be an effective forum to mutually agree on the list of preparatory activities that can be accomplished.

Our team completed 90% of all preparatory activities such as script writing, design and development of test framework, etc. during the first three iterations. However the test environment was not stable enough for our team to progress further. This is when we collaboratively decided to spend the next iteration in stabilizing the test environment. We deployed the entire team in validating the test environment and finding as many issues so that the IT team resolves all issues and provides us a stable environment.

We implemented iteration reviews and retrospectives in order to review progress, collect feedback and validate assumptions at regular intervals.

C. Iteration Reviews and Retrospectives

We conducted a review meeting per iteration at our customer’s location. We invited all stakeholders including representatives from senior management such as Sales Director and Delivery Manager to these meetings. All our team members participated in these meetings. We used Microsoft...
PowerPoint to present the status of our project along with outstanding issues and risks.

Through these meetings we could facilitate collaborative discussions, understand the changes in customer expectations and arrive at collective decision in order to accomplish project goals. We discussed unresolved technical as well as operational issues in order to identify action plans for timely resolution. Also we identified new risks and arrived at risk mitigation plans. Besides, in some of these meetings we reemphasized the high level vision of the project to everyone in the team.

7. PHASE III – TRAINING AND CLOSURE

The objective of this phase was to provide training on our Performance Testing framework to our customer’s team members so that they gain confidence in using our framework and understand the nuances of analyzing test results. This phase included two major activities – a) Knowledge Transfer on the design and the code base of our framework, and b) Training on analyzing and compiling test results in order to create test reports. We completed this phase with a schedule overrun of 20% because of the issues related to the availability of participants.

8. CHALLENGES ENCOUNTERED

The challenges faced in this project were predominantly related to Phase II. These were technical, operational and personnel challenges. This section discusses some of the critical challenges we encountered in this project. Per our estimates, the net impact of these challenges could have caused more than 70% of schedule and cost overruns. However we could minimize schedule variance through customer collaboration and course correction discussed in the next section.

A. Unstable Test Environment

According to our initial plan we had made an assumption that a stable test environment would be made available by the end of the first iteration. However, when we started the second iteration we faced several technical issues that indicated that the test environment was not stable. We spent the second and third iterations to learn the problems and realized the need for additional efforts to stabilize the test environment. Though we collaboratively decided to spend the next iteration in stabilizing the test environment the overall impact was a schedule slippage of more than 6 weeks.

B. Unexpected Outages

In Performance Testing every test cycle consumes several hours of test runs. In our case we had tests that required more than 24 hours of execution time. For this, we required network connectivity as well as the test environment available all the time without any interruption. However, we had few unexpected network as well as server outages that aborted some of the tests. Subsequently, we had to repeat those tests by setting up the test bed and executing them. This resulted in a negative impact on task completion. We collaborated with our customer in running some of the tests over weekends with less manual intervention in order to minimize schedule overruns caused by such outages. However such measures did not nullify the schedule slippage caused because of network and server outages.

C. New Builds

The objective of this project was to understand the performance characteristics of the product under test and suggest optimal configuration parameters. To accomplish this objective, we required a stable build. However, during test execution we came across critical defects that needed to be fixed. Hence, we had to wait until a new build arrived with bug fixes. This delayed our project further by two weeks.

D. Changes in Team Members

At the end of Phase I, we had to involve a new Database Expert from one of our facilities in India as the Database Expert in United States had to be released because of critical reasons. The newly inducted Database Expert in India was not collocated with the project team. Also, two of our team members had planned vacations at the end of the project. As the project underwent schedule slippage, we needed the team intact for additional weeks. However, team members who had planned their wedding or other such personal events could not change their vacation plans. In order to manage this situation we decided to introduce two new team members. Eventually, because of such changes in team, we had to budget for additional efforts in order to facilitate Knowledge Transfer to new team members.

9. IMPACT REVIEW AND COURSE CORRECTION

We had meetings with project team members to perform Root Cause Analysis [6, 7] and identified root causes as well as preventive and corrective actions. Subsequently, we scheduled a special review meeting at the end of the twelfth
iteration with all stakeholders to review the overall impact of some of these challenges. This meeting provided us a forum for constructive discussions because of the following reasons.

a) All project team members as well as other stakeholders were aware of the project happenings in detail.

b) All decision makers were technically savvy and appreciated the issues.

c) Due to the implementation of agile practices we did not have any last minute surprise.

Our customer had schedule pressure. Meanwhile, the project was facing both schedule and cost overruns. We had multiple rounds of group discussions to identify various options for managing such a tight situation in this Fixed Price project that started with fixed-scope, fixed-schedule and fixed-cost. We narrowed down our discussions to select one of the following options.

1. Negotiate a change request to compensate for all delays and continue the project.

2. Reduce the scope by analyzing all pending requirements and optimize the cost of change request.

3. Pause the project and restart it in Time & Material model.

Eventually, we decided on the second option. Our customer agreed to reduce the scope by removing a set of redundant test cases as well as low-priority test cases with no impact on project goals. This decision helped us reduce about 25% of the schedule slippage. Besides, the challenges related to the stability of test environment and builds were considered by our customer and we mutually agreed to raise a change request in order to balance the impact caused. This created a win-win proposition to deliver this project.

10. BEST PRACTICES

We identified three key best practices in this project. One of them was very effective in making a practical proposal and the other two proved to be very useful in Project and Customer Management. This section provides a brief description of these three best practices.

A. Phased Approach

Proposing a phased approach helped us understand the project well and take the next steps. Without this, we would not have come up with refined estimates for this project. Also, phased approach helped us demonstrate our experience and expertise during the first phase in front to all stakeholders in a systematic manner for further budget approvals.

B. Implementation of Agile Practices

Besides proposing an iterative approach for Phase II, we decided to introduce agile practices that helped us respond to changes as well as collaborate with our customer. Without agile practices we would not have gained enough confidence and trust in working as a team. This facilitated constructive discussions with customer when we wanted to present project challenges and overall impact in order to seek course correction.

C. Transparency and Quantitative Management

We maintained transparency in the team by means of sharing project tasks, schedule, issues, risks and queries. Also, we practiced quantitative management and reported metrics such as Schedule Variance, Effort Variance in addition to Product Quality Metrics. With this approach we established a common understanding across stakeholders on the overall status of this project. This increased the level of trust in the team.

11.LESSONS LEARNED

At the end of this project we realized two critical factors that could have helped us in managing this project better - a) a separate phase to validate the test environment and b) an additional environment for bug verification. We derived two important lessons based on these factors. This section provides a brief description of these two lessons.

A. Validate the Need for Additional Phases

The first lesson we learned in this project is about the need for additional phases. Initially, we did not plan for ‘Environment Stabilization Phase’. We had made an assumption that the test environment would be stable and available by the end of the first iteration of Phase II. However, we had to spend efforts in stabilizing the environment and reprioritize our tasks. Performance Testing projects have long test cycles, and long-running tests and hence they need stable environment. Besides, load simulations and stress tests require adequate infrastructure capacity as well as reliability.

B. Validate the Need for Additional Environment

We came across several functional issues during the initial iterations. In order to debug these issues we had not budgeted for a separate environment. We faced delays in getting servers on need basis for debugging. These servers were not available on a permanent basis. In this context,
the second lesson we learned in this project is about the need for additional environments. A debugging environment would have helped us parallelize debugging activities and provided us some leverage to control schedule overrun.

12. FIXED PRICE AND DISTRIBUTED AGILE: FINDINGS

The need to execute Fixed Price projects prevails because customers prefer to get projects done in fixed-schedule and fixed-cost. However, there are several factors that come into play while making decisions to execute projects in Fixed Price model. Some of the key factors include Requirement Stability, Maturity of Technology, Estimation Process, Project Management Processes, Customer Collaboration and Proposed Approach. We scheduled this project over an elapsed time of six months and eventually delivered it in eight months. We had several challenges in executing this project in a Fixed Price model. This project involved several Software Engineering activities such as Requirements Gathering, Framework Design, Framework Development, Framework Testing, Test Data Generation, Test Bed Setup, Performance Testing, and Test Reporting. While working with our project team on all these activities, we came across several Findings. The following are the key Findings that can be of value to practitioners.

1) Scope stability is very crucial to executing Fixed Price Agile Projects. This means that there has to be adequate clarity on what is included in scope and what is not. An open-ended scope definition does not suit Fixed Price engagements.

2) Provision for scope adjustment is as crucial as the need for scope stability. Software projects succumb to several types of changes such as changes related to requirements, technology, infrastructure, and people. In practice, executing a Fixed Price engagement with fixed-scope, fixed-schedule and fixed-cost is attempted with some form of contingency building in project schedule. Such contingency measures need to be communicated to all stakeholders. This creates transparency and trust. Besides, some provision for scope adjustment is required to handle situations related to schedule slippage or cost overruns.

3) In case of Fixed Price Agile Projects, doing multiple small projects is better than doing a single big project. One way to accomplish this is to divide and conquer. This can be implemented by means of phased approach.

4) Fixed Price model seems to work better in agile ecosystems than in traditional ecosystems. Agile ecosystems provide for a greater level of collaboration and commitment. This creates mutual understanding and trust. Implementation of agile practices was one of the driving factors that helped us complete this project successfully.

5) Combining Fixed Price and Distributed Agile can bring in benefits in terms of providing visibility and predictability at regular intervals. However, such a combination may not yield productive results when it is used only to introduce new requirements during project execution.

6) Fixed Price projects require lot of transparency without which reviews and negotiations will not provide win-win results. Planned reviews conducted in systematic manner with project performance metrics and quality metrics establish transparency and confidence.

7) In order to execute projects of this nature, it is very necessary to have stakeholders who are technically savvy and are capable of understanding and appreciating technical issues. This means that the Governance Model of Fixed Price projects need to have senior members who can understand and appreciate project happenings in terms of progress, risks, as well as technical issues. Such senior members need to have influencing and decision making skills in order to find practical resolution to problems.

8) Customer Experience Survey, an independent survey conducted every year by one of the corporate functions of our organization, revealed high level of customer confidence in our technical expertise, and management support in completing this project.

13. CONCLUSION

Fixed Price Distributed Agile projects are very challenging to conceptualize, strategize, plan and execute. The number of research studies and experience reports available on this subject is very limited. In business environments, the need for Fixed Price proposals to deliver software services will continue to grow. The Best Practices, Lessons Learned and Findings shared in this paper are based on our experience in a specific project. However they address common themes that are applicable to software projects in general. Keeping in mind the time-to-market and budget constraints as well as the benefits of iterative lifecycle models, practitioners need to choose the right set of best practices that enable successful completion of Fixed Price projects.
REFERENCES


