



Managing Offshore Software Projects The Agile way

Thavaranjan Thavendran

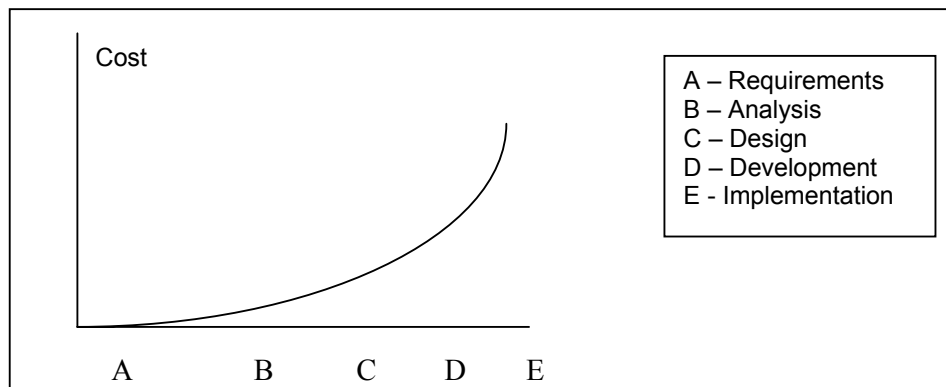
Previous Article

This is part two of a two article on Managing offshore projects using agile techniques. In the previous article, the author covered many of the issues surrounding the situation and started discussion on what could be done to make it a success. This article continues the discussion.

Agile Based Testing

This is one of the most important issues in the software development process. No matter how technically brilliant your company is, if the products that you built don't function or breaks down when put to use in the real environment, the reputation and the image of the company will be at stake. End-users evaluate your work based on how well your software performs and meets their expectations. They do not evaluate you based on how much effort or the complicated piece of coding you have put into the software.

In order to prevent your software from breaking down, significant investment has to be made in testing and quality assurance. Testing and bug fixing can be a very expensive process; therefore early detection can save in a great way. The graph below shows how costs increase as you move through the software life cycle.



The methods described below provide a practical approach towards early detection and prevention of issues.

Test First Development

“Test First” is all about creating tests cases before the coding starts. This makes the coding process easier and faster. It helps developers to understand the requirements clearly thereby guiding the development of code via test cases, and provides a great sense of confidence to move forward.

The “Test First” approach also helps reduce misunderstanding of requirements, provides clear indication of progress and helps control scope creeps. One of the other interesting points with this approach, is that, some developers always feel bored and not motivated to write test cases after coding. This is due to their over confident nature where they assume that everything will work, so why bother writing test cases. Due to this they release their modules without much testing.

Now how do you use this concept? With this concept you shouldn't write any piece of production code, till you have got your failing tests in place. The software will pass only when you write the relevant coding for it. It is not necessary that you should write all test cases first and then start on coding. Some prefer to have all the test cases written first and then work on the coding to make each test case pass; some others find it easy to write one test case, and they write code to make it pass. They then move to the second test case, which follows by coding for that test case and not anything more. Therefore this continues in a loop till all test cases are implemented. In order to make this approach even more beneficial, test cases have to be written even when attempting to fix a bug, not merely when implementing a new feature.

To successfully implement this concept there are some pretty good open source tools, which can be made use of, such as NUnit and JUnit, in which you can assign your test cases and run the tool after implementing each piece of code. When a test case passes, these tools have very good ways of indicating success or failure.

Peer Reviews

Peer reviews are all about getting an equal - developer peer - to go through ones work. Traditionally peer reviews were carried out merely for evaluating code quality, but nowadays they are used to evaluate many other factors. Some of these include professionalism, consistency, standards, and cleanliness in code, requirements, design, estimations and documentations. Some management tends to feel that this is an unnecessary overhead, but if this is properly undertaken, it could yield very good results.

Although peer review has its positive sides, there can also be resistance among developers. Every developer takes pride in the work they do, and much of the self worth is tied down to their own work. This prevents them from admitting any mistakes they might have made in the code, or ask for help from their peer to review and point out their mistakes due to ego. Therefore peer review goes hand in hand with the human nature of ego, and has to be managed to obtain better outcomes. Without proper management, this could sometimes lead to hard feelings between the developer and the reviewer as well.

In order to overcome the above problems, it would be best if the concept of pair programming is used. Pair programming is all about working in pairs, in order to create a trusted peer. Pair programming can increase productivity of both individuals rather than the common notion that productivity will drop. A thorough study has been done by Laurie Williams and Alistair Cockburn in the paper “The costs and benefits of pair programming”.

Manual and Automated Testing

In order to cope up with today's aggressive demands in time-to-market requirements, application development has to make sure that it's always kept on the fast track. Automated testing tools provide a very good approach towards speedy verification and testing of your work. They do this by allowing repeatable and predictable method

of continuously testing your application functionality, and thereby eliminating the need for tying up large numbers of expensive resources.

There are several automated testing tools, such as:

- NUnit
- Junit
- TestDriven
- ZaneBug

which can do a very good job of automating your test process. In order to make good use of these tools, developer's should develop a different perception with regards to coding. They initially (or at least after the basic coding is done and before fine tuning and bug fixing is carried out) are made to identify all possible test scenarios/cases for a given task or module. These test cases should be an integrated part of the module specification for any module in a project.

Most of these test cases will be converted to unit tests during development. Unit test can easily test almost 90% of all aspects of a software application, with some exceptions on a few graphical components. If this is properly done, it can reduce your manual testing time by 50 – 60%, which is a huge cost saving. Unit tests have to be considered as part and parcel of any development, therefore when estimating a module, time should be included for this as well.

A module should not be assumed completed till these cases are also developed. Due to recursive nature of unit tests, they are very effective in handling faster software release cycles, and also help a great deal in detecting any broken codes. Unit tests can also help new project members to do changes in existing code with some assurance that any broken code can be easily identified.

In most of the recent projects I have managed, we had cruise control running on a daily basis to build the software and run the unit test cases every night. The results of this build are emailed and are available for the developers to review, when they start work the next day. In the project plan I have allocated some room (30 minutes to 1 hour) to attend to these issues on a daily basis.

End-user Testing

End user testing should happen at the end of each time box, and acceptance should be obtained. I would like to refer to this end user testing at each time box as the “mini acceptance testing”. Sometimes it might not be possible to obtain a formal acceptance due to the incomplete nature of the software; some clients may have the fear that up coming development on other modules might have an impact. It is not necessary for a formal sign off, instead it can merely be an email stating the acceptance, although formal signing off is mandatory at the final completion of the project.

Before releasing the software to the client for mini acceptance testing, proper testing with an end user point of view has to be carried out to ensure:

- Robustness
- Stability
- User friendliness.

These criteria are very important concerns to win the hearts of the clients. Also the more the software is user friendly, the easier it will be on user training. One of the key things that can contribute to user friendliness; is to adapt the behavior and look and feel of an environment to one with which the end users are already familiar such as “Windows”. You can thereby avoid re-inventing the wheel all over again.

Progress measurements on testing tasks is vital. Without proper methods for measuring progress, testing can go out of control, since testing is an infinite area. Each tester can come up with different numbers of test cases for the same software. In most cases, the known test cases defined upfront do not constitute all possible cases. Most undefined cases are brought to light during hands on testing. Therefore measuring the progress becomes a difficult task. In fact these types of ad-hoc testing are not measurable at all, therefore the ideal way would be to define a period for ad-hoc testing.

If for instance, assuming your client asks for the amount of testing you have done so far, what would your answer be? The answer to the above question is subjective to the different dimensions, you would consider in measuring.

Offshore Change Management

It is a known fact that requirements will evolve throughout the project, for several reasons. People change their minds quite often. For instance when software is given for testing they might suddenly realize that they have missed some functionality. Some requirement might have not been implemented the way they had in mind. Changes in governmental legislation, or some competitor is providing more functionalities than what is being offered.

There is no point trying to put a full stop to changes. Doing this will only make you deliver software which the customers thought they initially wanted. It will not be a product which they can use successfully now. Therefore one thing is sure in any project, and that is, changes. Agile is quite flexible in this nature.

In the initial stages of the project, requirements analysis documentation is done just sufficiently to pinpoint the scope, build a high-level schedule and develop a project plan. Not much time is invested on detailed requirements documentation, which we know for sure will be a waste. In order to maintain the flexibility, the entire project is broken down into modules and allocated into separate time boxes. Detailed requirements analysis and documentation is carried out throughout the project in each time box, which minimizes changes to a great extent. Although we try to manage changes this way, it doesn't mean that changes are prevented. At each time box when detailed requirements analysis is done, the requirements are frozen, in order to provide stability for the development phase in the next time box.

Success factors for Offshore Development

Dedicated person from customer

Having a dedicated person at the client end helps significantly in speeding up the project. Some companies have the mentality that dedicated people will only be a waste. This impacts the project in a major way by making it harder to achieve set deadlines. A dedicated person with the know-how and authority is very important at the customer site.

Client Involvement

Some clients are of the mentality that, when a project is awarded to a software vendor, it is the software vendor's problem and not theirs. The vendor has to solve all problems without troubling the client.

After requirements gathering and approving requirements specifications, it would be best if the client prepares the user acceptance plan in advance, which can be used by the development team to understand the requirements more clearly. Clear understanding of the requirements can provide a good start and an excellent foundation for the software to be built.

Need for an Offshore Project Manager

Many projects fail not predominantly because of technical skills, but because of poor or lack of proper project management. It is very important for the offshore software team to have a dedicated project manager. Through observations, some clients opt to do the project management themselves remotely. Often these projects have failed miserably and have caused good business relationships to break.

Not having an offshore project manager can cause many other issues, such as

- Not achieving the cost benefit in off shoring as originally planned.
- More time required, especially with the time zone difference. This also adds to the increased cost, due to the fact that time is money.
- It might very often be required for the client project manager to travel to the offshore destination. This can further add to the cost issues, with travel cost (flight cost and local travelling), cost for accommodation, daily allowances, and travel insurances having to be paid on top of the usual salaries.
- Cultural Issues. Imagine an American managing a team of Indians or Sri Lankan, whose culture is totally different to that of an American.

How does Agile relate to Fixed Price Contracts?

When I talk about agile development, I have often heard people asking me whether agile works for fixed price contracts. Usually when people say fixed price contracts they mean fixed price, time and scope. This means you require stable and solid requirements. I have tried a couple of approaches here, and both worked really well for me.

The first approach is for projects which fall into the domain we specialize in. For these types of projects, it is a must that you have the following;

- A known team
- Known technology

After making sure of the above, then I start working on the project contract. This document defines scope, features, deliverables, timelines, and the price for the project, which forms the working relationship between the customer and the software vendor. Pretty much everything documented here is fixed.

It is very important that as project managers we be tough on the agreed deliverables, which means change requests are not entertained. Instead adopt the "exchange requests" concept. So if there is a change request, then negotiate with the customer to find out which module you can take out to accommodate the new request. The module that you take out should be of an equal weight to the module that you are putting in.

The second approach is for projects which are not in your domain. In this scenario the approach that has worked for me is to break the project into two sub projects. The first sub project will focus on producing a detail requirements analysis, documentation and estimation. This will provide you with sufficient information to determine the costs. Based on this a new contract could be formed to initiate the second sub project, which will focus on implementation.

How to Select an Offshore Vendor

Selecting an offshore vendor can be a challenging task. Numerous factors will have to be considered in this process. The following are some factors which I have gathered from our potential clients.

Stability

How would it feel if while your project is being developed or maintained, your offshore company collapses? All your investment would be a waste, and you might be left with a product with no support for it. It might be possible to find some other software vendor to support it, but imagine the painful process you would have to go through to find that vendor, and the steep learning curve they will need to come up to speed in maintaining and supporting your product. For this reason it is very critical to choose the right company.

The following are some factors which could give you the answer that you are looking with regards to the question on stability.

- Past few year audited annual report
- Shareholders and their strength in supporting the company
- Track record of the person running the company (i.e. General Manager, CEO, etc)
- Company's past project performance (case studies on completed projects), and current clientele
- Employee turnover rate
- Political situation of the country

Size and Infrastructure

Some also consider the strength/size (number of employees) of the company as key factor towards choosing an offshore partner. From my observations, the size of the company should not be a deciding factor. What's important is the quality of the people who'll be involved in the project, stability of the company and the reputation it holds locally or globally.

Due to the distance, project coordination and control becomes a real challenge in a context where effective communication is one of the critical success factors towards achieving a successful offshore endeavour. Many companies nowadays use email, chat, video conferencing, pc-to-pc phones, etc for communication. As well, they use other tools such as ftp, and http for file/application sharing and document repository. Careful attention has to be paid to telecommunication infrastructure.

Furthermore you might want to look into the type of machines, software and network used for development. A low-end machine would take more time to compile than a high-end machine. This means a task, which should have finished in 10 hours, takes 12 hours to complete because of poorly performing machines. This also means you will be either paying a high price, or be delayed with the product.

Communication

Strong communication is crucial for the success of offshore projects, and therefore language is a common concern among many. Certainly many companies would prefer to do business in their own language. Most business communication in Asian countries is carried out in English. Although English is spoken throughout the world, accent/slang and body language varies from country to country; which might lead to misunderstandings.

Due to this situation many companies move towards formalizing their communication by means of detailed documentations and formal processes. This seems to help to a great extent especially with geographically dispersed teams, but make sure that you do not over do this.

Employee skill set

The majority of companies will have skills in most common programming languages, such as C#, VB.Net, Java, etc. How experienced they are however, can only be determined by talking to them. Some software vendors will allow you to interview the team they will put together for your project, or might show you their resumes. Take this opportunity to identify your team.

Some offshore projects might require specialized domain knowledge, like banking, finance, engineering etc. It would be highly advantageous if your vendor had prior experience in this field. Some vendors would claim, although they don't have this knowledge, that they still can do the job. Be cautious when doing projects with these sorts of companies.

Quality Standards

CMM (Capability Maturity Model), SEI and ISO are widely known quality standards, and you would notice some companies claiming that they are CMM or ISO certified, mainly to differentiate them from others.

What these certifications mean is that the firm understands the quality assurance concepts and has gone through a process of implementing it throughout the company. This does not necessarily mean that they practice what they preach. Possessing a certification doesn't guarantee that the company will deliver high quality software.

It would be best if judgment to select a vendor is made based on:

- Prior completed projects
- Demonstration of the process along with real evidences (such as documents made for the projects, i.e. project plan, status reports, requirements/functional specifications, etc)
- Communicating with another client if possible.

Conclusion

Globalization is here to stay and the practice of offshoring will increase on a daily basis. A venture into offshoring is always a risky matter; however it is also quite competitive. Agile methodology can help mitigate many risks posed by offshoring, such as culture, trust, communication, transparency, ROI, dispersed locations, time zones and the constant change in requirements. This white paper discusses how agile methodology can be used in an offshoring context to manage the risks.

References

Agile Manifesto, “*Manifesto for Agile Software Development*”, www.agilemanifesto.org

“*Code the Unit Test First*”, <http://www.extremeprogramming.org/rules/testfirst.html>

Kevin Aguano, “*Managing Agile Projects*”, *Multi-media publications*, 2005

Laurie Williams and Alistair Cockburn, “*The costs and benefits of pair programming*”, <http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF>

Martin Fowler, “*The New Methodology*”, www.martinfowler.com/articles/newMethodology.html

Pekka Abrahamsson, Outi Salo, Jusi Ronkainen, and Juhani Wartsa, “*Agile software development methods*”, www.inf.vtt.fi/pdf/publications/2002/P478.pdf

Stephanie Moore and Liz Barnett, “*Offshore outsourcing and Agile Development*”, www.forrester.com/Research/Document/Excerpt/0,7211,34978,00.html, 2004

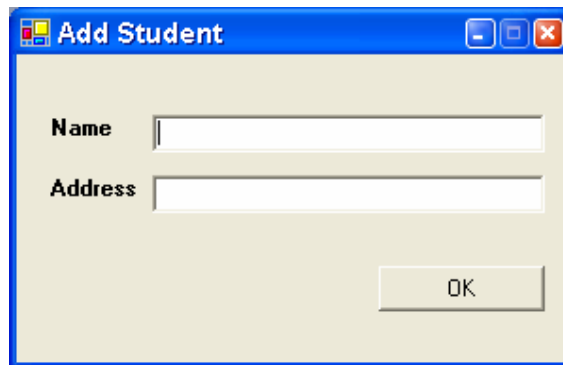
Appendix A

This is a sample functional specification made for a module which adds students. This template provides some sample data to explain each section.

M1: Add Students

Date	Initials	Description
21-06-2006	RT	First version - ready for approval
22-06-2006	KT	Approved
23-06-2006	RT	Replaced the screen shot to accommodate the address field

Screen Layout



Screen 1

This screen allows school administrators to add new students to the system.

Field Definitions

Field	Type	Enabled	Description
Name	Label [25]	Yes	Name of the selected Student. If the student name is too long to fit, it will be truncated and appended with "..."
Address	Label [40]	Yes	

Actions

Action Element	Enabled	Action
Pressing the 'OK' button	Yes	Insert the Student to the database and close the form

Screen Navigation

Screen	Activated By
Screen 1	M2 – View Existing Students

Test Plan

ID	Action	Expected Result
M1-1	Press 'OK' without entering any data	Error message should popup to warn the user to enter data
M1-2		

About the Author

systems integration.

Thavarajan Thavendran has engaged in management of several offshore and onshore projects as a Project Manager. He has managed projects for many clients around the world, including Norway, Denmark, Australia, US, Jordan, Sri Lanka and Dubai. He has experience managing projects from a wide range of industries, with specialization in automation of stock exchanges and offshore project management. The types of projects he has managed are client-server application, web application, database applications and

Thavarajan has over 9 years experience in the IT field, and holds a BSc.(Hons.) in Information systems (from the Manchester Metropolitan University, UK) and MBA from the University of Lincoln, UK, and is also an ISO certified internal quality auditor. Author's Email: ranjanthavendran@yahoo.com

Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called “Method H”™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au