

Title of Paper:

Making Agile Work in Offshore Outsourced Model

Submitted for:

The 6th Annual International Project Management Leadership Conference [PML 2011]

Authors:

Amit Gupta, amit.gupta@hsc.com

Hemanta Chandra Bhatt, hemanta.bhatt@hsc.com

Organization:

Hughes Systique Corporation, Gurgaon, INDIA

MAKING AGILE WORK IN OFFSHORE OUTSOURCED MODEL

Amit Gupta (amit.gupta@hsc.com)

Hemanta Chandra Bhatt (hemanta.bhatt@hsc.com)

Hughes Systique Corporation, Gurgaon, INDIA

Abstract

Other than understanding the organizational and cultural aspects related to application of Agile methods, modification and adaptation of Agile practices to suit the organization's business context and existing systems and processes is imperative for making Agile work. For companies in the Indian software and IT services sector, these challenges are compounded due to the offshore outsourcing execution model coupled with the increasing trend of fixed price contracting. Fixed Price Model presents additional set of constraints that may make application of Agile while achieving margin targets a difficult goal to achieve. Fixed price for changing scope runs counter to profitability and fixed scope is not considered to be a good example for application of Agile.

This paper is about making Agile work in offshore outsourcing model whereby the focus is on Scrum method which is arguably the most popular Agile method and has been used by many organizations for building commercial-grade software. This paper describes the pure (or standard) Agile Scrum practices, explores the challenges in making pure Agile Scrum work in offshore outsourcing model and proposes modifications and adaptations (including to the management style and practices) to overcome those challenges especially in case of fixed price projects. The proposed modifications and adaptations are based on real life experience gained while working on Agile Scrum projects in Hughes Systique Corporation (HSC).

1. Introduction

SDLC models like waterfall (including its variants), iterative, incremental, V, etc. have been around for last so many decades for software development. Agile [1] methods like Scrum [2], XP, FDD, etc. have got added to the above list since the last decade. Agile methods claim to be adaptive in nature and are based on the principles enlisted in the Agile Manifesto [3].

Traditional or pure waterfall (depicted in *Figure 1*) has multiple phases which are executed in a sequential manner. Some of the key characteristics of traditional waterfall are as given below:

- No flexibility which means "do it right first time"
- Assumes 100% visibility at the start, everyone knows what is expected

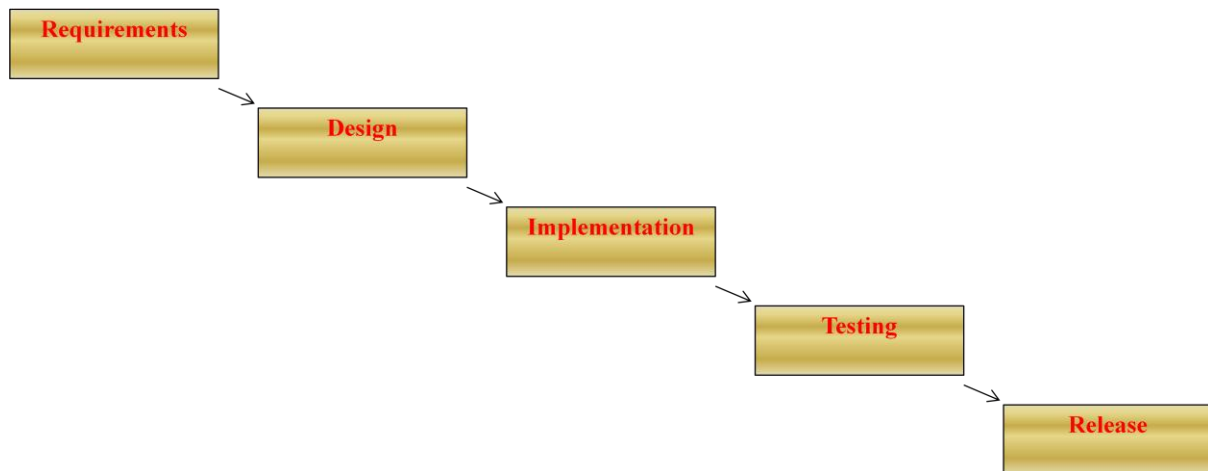


Figure 1 Traditional or Pure Waterfall

In reality, however, pure waterfall is never used due to reasons some of which are given below:

- Business case and customer expectations change
- It is extremely difficult to do it right first time, visibility is 100% only at the end!

The “real” or modified waterfall (depicted in *Figure 2*) has the following characteristics:

- Revisits to the past phases (changes)
- Multiple back and forth iterations amongst phases (defect seepage, rework)

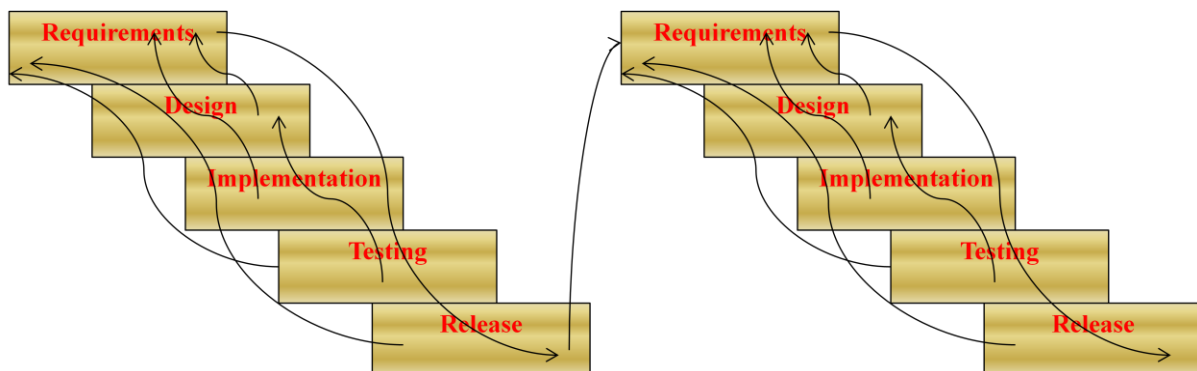


Figure 2 “Real” or Modified Waterfall

2. Agile

2.1 Agile Methodology

Agile methods are supposedly better at handling changes during software development as they believe in “responding to change over following a plan” [3]. Some of the key characteristics of Agile methods (depicted in *Figure 3*) include:

- Release in multiple increments (sprints) to get continuous feedback and evolve product as per market needs
- Development in multiple back and forth iterations amongst phases within a sprint
- Activities of various phases performed in parallel

- Continuous development and testing (this presents a challenge in terms of differentiating original work versus rework)

Though Agile methods, as depicted in *Figure 3* below, look similar to modified waterfall (*Figure 2*) at a high level, the key difference is the duration of each iteration and the parallelization of phases within individual iterations.

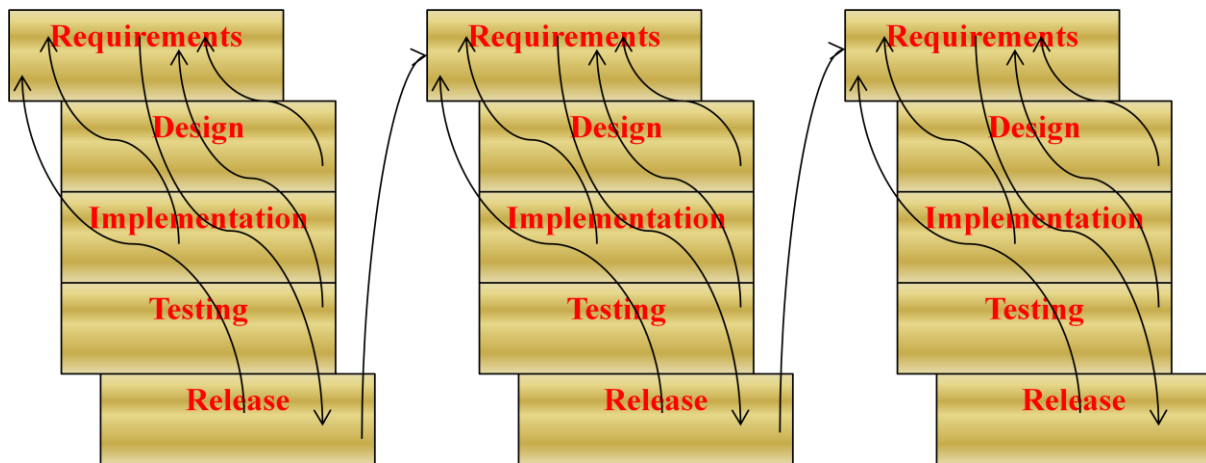


Figure 3 Agile Method

Agile methods work well for small co-located teams facing rapidly changing conditions. These methods can benefit greatly in the following situations:

- Customer not sure of exact requirements
 - Investing in new or innovative product concept to tap certain market need
 - Trying to create a new market segment
 - Seeking to create technology or product leadership where time to market is critical
- Project team not sure of exact design and implementation approach
 - Using new or emerging technologies and frameworks
 - Working on a new or innovative product concept

2.2 Agile Scrum

Scrum is arguably the most popular [4] Agile method and has been used by many organizations for building commercial-grade software. Scrum has found increasing acceptance in the companies in the Indian software and IT services sector for following reasons:

1. Scrum offers definitive practices that can be applied consistently across various projects.
2. Scrum has been observed to work well within the CMMI implementation [5], [6], [7], [8] and hence can leverage and strengthen the CMMI achievements of the Indian software and IT services companies.

3. Offshore Outsourcing Model

Outsourcing is prevalent since last so many decades. After the Y2K bug, offshore outsourcing of software projects to locations in far away countries like India, etc. has become an important part of the business strategy of many companies, especially in the USA, due to the following reasons:

1. Skills at reduced cost arising out of labor arbitrage. This is influenced by differences amongst various countries in respect of macro-economic factors such as currency exchange rates, cost of living indices, purchasing power parity, tax policy, etc.
2. Work beyond business hours coverage due to time zone differences. This is influenced by geographical factors.
3. High level of confidence in respect of cost-effective, timely and high-quality project execution. This is influenced by a combination of factors such as technical and management competencies, systems on processes developed based on best practice frameworks like ISO 9001, ISO 27001, CMMI, etc.

The offshore outsourcing execution model is a distributed delivery model which has multiple stakeholders as depicted in *Figure 4* below.



Figure 4 Offshore Outsourcing Model

4. Fixed Price Development

The offshore outsourcing model has matured over last so many years resulting in offshore software companies moving up the value chain.

A major trend seen in the past few years in the outsourcing model is the increasing prevalence of fixed price projects [9]. Business managers are attempting to move away from effort-based contracting (time and material) to outcome-based contracting (fixed price). This provides customers definitive figures for their software budget and helps in controlling the IT budget by passing on the risk of inflated costs to the software partner.

5. Making Agile Work in Offshore Outsourcing

5.1 Agile and Offshore Outsourcing

The challenges in using Agile in an offshore outsourcing delivery model are compounded due to the impact of individual challenges with using Agile and offshore outsourcing. The list below summarizes the major challenges:

1. Agile recommends co-located teams which is not possible in case of offshore outsourcing or distributed team.
2. Agile requires direct and close interactions with customers which is generally not the case in the offshore outsourcing model where the onsite team or onsite coordinator (maybe from the offshore team only) is sandwiched between the customer and the offshore team.
3. Agile requires close interaction with the customer, even from business teams e.g. customers to test the delivered software and provide feedback to the technical team after every release (typically after every 2-4 weeks) where as water fall model does not need such a close connection

For making agile work in offshore outsourcing the multiple stakeholders (as depicted in *Figure 4*) need to carry certain responsibilities.

Customer	<ul style="list-style-type: none"> • Allocate needed time for interaction with project team • Test and provide feedback as soon as sprint release is made • Communicate requirements, changes as soon as possible
Offshore Team	<ul style="list-style-type: none"> • Good understanding of business context of the software • High level of technical expertise, reasonably experienced • Self-organized, self-driven and self-motivated
Offshore Organization	<ul style="list-style-type: none"> • Strong retention measures and transition-out (attrition) • Rigorous tracking of cost, profit (scope creep in fixed price Agile) • Strategy for "profitable engineering" (risk-reward sharing)
Onshore Team	<ul style="list-style-type: none"> • Allow participation of offshore team in customer meetings • Ensure customers can approach offshore team if needed • Ensure latest information is available to offshore team through common tools for CM, testing, bug tracking, etc.
Onshore Organization	<ul style="list-style-type: none"> • Establish well-defined Agile process for working with offshore partners • Establish well-defined Agile process for working in distributed team environment
Other Distributed Team(s)	<ul style="list-style-type: none"> • Use formal communication mechanisms in addition to Agile way (frequent verbal, face-to-face interactions)

Table 1 Responsibility of Stakeholders for Making Agile Work in an Offshore Outsourcing Model

Other than understanding the organizational and cultural aspects related to application of Agile methods, modification and adaptation of Agile practices to suit the organization's business context and existing systems and processes is imperative for making Agile work. For companies in the Indian software and IT services sector these challenges are inherent due to the offshore outsourcing execution model they use.

5.2 Agile in Offshore Outsourcing with Fixed Price

The challenge of using agile in offshore outsourcing is further compounded due to the increasing trend of fixed price contracting which presents additional set of constraints that may make application of agile while meeting agreed contracts and thus achieving margin targets a difficult goal to achieve. This challenge is inherent to Agile method as Agile philosophy is about embracing change which might lead to scope creep thus putting pressure on the initial estimates, approved plans and thus profit margins. Fixed price for changing scope runs counter to profitability and fixed scope is not considered to be a good example for application of Agile.

Though fixed price model suits the product and business managers from budget management perspective, but they need to specify the requirements clearly upfront for the software partner to quote a fixed price. This poses contracting challenges – fixed price implies fixed scope (complete and clear requirements upfront) which is a contradiction to the Agile philosophy where the requirements evolves during the project execution through internal and customer feedback on intermediate releases.

The core issue around Agile in offshore outsourcing with fixed price is measurement and tracking of the scope of the work.

Following approaches are commonly employed in this scenario:

- In the first approach the commercial negotiations for the initial work order are carried out based on the requirements available at the initial stages of the project. These requirements are tested and necessary additions, modification and deletions are done as the product evolves through change orders. Change orders detail out the commercial impact which is used to revise the commercials. The change orders may necessitate discussions and disagreements between the customer and the software development partner and may impact the project's progress and successful completion. The negotiations for change orders may also require involvement of those in the organization who may not be directly involved in the day to day software development like representative from sales & marketing, engineering management and finance.
- The second approach involves requirements trade-offs throughout the project's execution. In this case the initial commercials are done for the complete project assuming that the upfront available requirements will hold true. Development of the features are started based on their priority and during execution product owner and the software development partner can trade-off the requirements i.e. new requirements can be exchanged with some existing requirement. This approach is better over the first approach in its ability to maintain the final cost within the budget. The assumption is that requirements will not change fundamentally and significantly and both the parties will tend to agree on exchanging requirements whenever such a situation arises.

6. Agile at HSC

6.1 Agile Experience at HSC

The following sections describe the experiences (including adaptations and modifications to pure Agile) gained at HSC while executing Agile projects in offshore outsourcing model with fixed price. HSC, part of the Hughes group of companies, is a communications consulting and software company. HSC executes large and small projects using different software development methodologies such as the Agile methods as well as the traditional waterfall lifecycle models.

The basic drawback in the second approach in Section 5.2 is that it assumes that a good amount of requirements are available upfront and there will be just a few delta changes during project execution. This assumption may hold true in cases where products are developed or maintained against clearly understood requirements and using proven and stable technologies. However in case of new and innovative products this may not hold true.

6.2 Agile Adaptations and Modifications

The table (*Table 2*) below lists the adaptations and modifications to the pure Agile scrum as were made while executing projects at HSC.

Area	Pure Agile Scrum	Modified Agile Scrum	Advantages of Modified Agile Scrum
<i>Contracting</i>	Commercial negotiation are done on the basis of defined work	Contracting is based on number of user points while the exact scope remains flexible Sprint 1 focuses on establishing relationships and processes for engaging the customer	User story point agreement gives space to both the parties to evolve actual requirements as product grows without any resistance at the operational level. Sprint 1 gives a high level idea about the end product and a fair idea about the weight of the release planned as part of this contractual agreement.
<i>Planning</i>	Sprint scope of work is limited to deliveries planned for that sprint	Sprint scope also includes design for the next sprint	Allows preemptive detection of technical challenges which ensures the team's velocity is not impacted by holdups due to technical challenges
	At the start of sprint in the sprint planning meeting	Done in advance of one sprint (min by 80%) and final closure at the start of sprint in the sprint planning meeting	Planning for a sprint in advance ensures planning is dynamic and the software development is always moving
<i>Execution</i>	All sprints deliver working	The first sprint is focused only on architecture and design.	Sprint focused on architecture provides end-game visibility and hence

	software	One of the sprint is focused only on fit and finish	better control throughout the project's life. Sprint focused on fit and finish ensures rough edges in installing and using the software are smoothed out. <i>Refer to Section 6.3 for details.</i>
<i>Tracking</i>	Through daily scrum meeting involving entire team Sprint review meeting	Through daily team meeting involving leads In addition to sprint review meeting there are program review meetings with customer and internal management	Helps manage more effectively especially in case of a large team Allows intervention in case of significant impact on the functionality (for customer) or financial health (for internal management)
<i>Change Management</i>	Changes are embraced	Overall scope is constrained in terms of number of user stories and changes are handled through requirement trade-off (replacing a requirement with another)	Ensures better cost management especially in case of fixed price agile projects
<i>Learning and Improvement</i>	Through sprint retrospective but is quite limited in nature	Through use of CMMI-based framework and extends to subsequent phases of the project and entire organization	Allows learning and improvement leading to higher productivity

Table 2 Comparative Analysis of Practices in Pure and Modified Agile Scrum

6.3 Modified Agile Scrum Process

The figures below depict the pure and modified Agile Scrum.

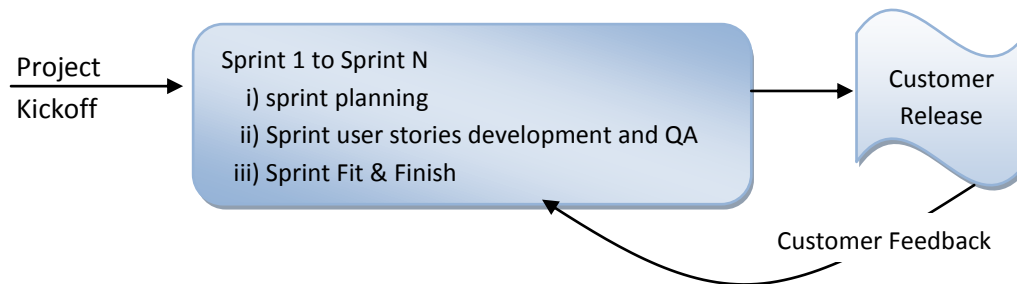


Figure 5 Pure Agile Scrum

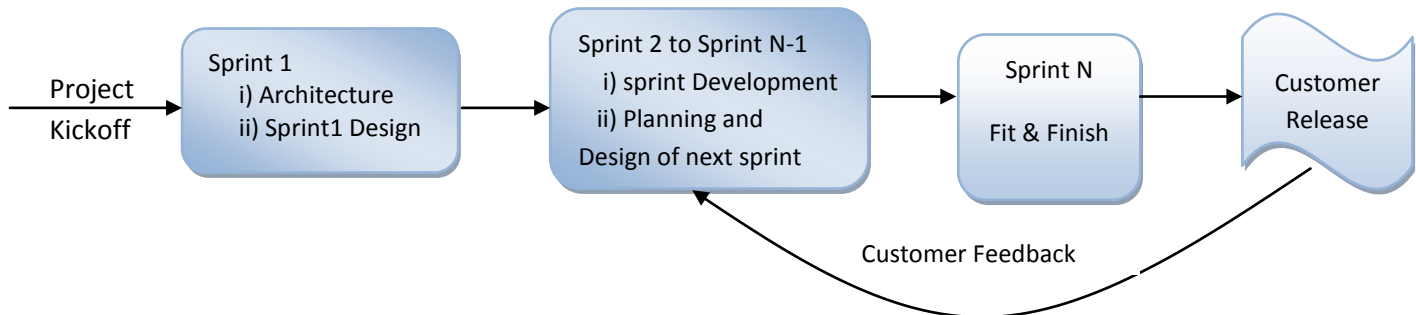


Figure 6 Modified Agile Scrum Cycle for Fixed Price Outsourced Project

Sprint 1 – Architecture Sprint

Agile treats all sprints as equal where the project team works upon selected user stories and delivers working software to the customer. In the modified approach, Sprint 1 is different as there is no working software that gets delivered. Instead Sprint 1 is focused on laying down overall product architecture, bringing the teams on both sides together and establishing the processes. This Sprint makes sure that everyone comes on the same page in relation to final product business idea and key requirements and gets connected which facilitate the coming sprints execution. This sprint also focuses on the planning and design for the next sprint.

If both the parties are working together for the first time or too new on the product, this sprint can be done in the mutual agreed model and the costing and thus commercial negotiation can be done after this sprint1. This one sprint gives a platform for the team to analyze the high level requirements, do some estimates for the total user points required for the current agreement and the associated process to give points to each user story.

Sprint 2 – Sprint N-1

In the modified approach these are regular sprints with certain modifications. In addition to the sprint's regular work the project team also works on the following activities:

1. Design and technical dependencies for the next sprint is worked on and closed. This is very important to keep the team fully engaged since as soon as they finish off one sprint they can pick up new stories for development
2. Sprint planning for the next to next sprint should be 80% done in this sprint, giving space for some last minute critical feedback incorporation and sprint planning closure for the next sprint. This should include the user story allocation for the sprint, sizing the Unit of work as User story points based on the processes established during the contractual stage.

These above approach ensure that the engineering team is productive and fully engaged when they are done with the current sprint. In absence of this, they may get hanged up due to dependency and design closure and may lead to delay in commitment, thus adding delay for the product owner and costing for the software development partner.

Sprint N – Fit and Finish Sprint

The focus of this sprint on fit and finish of the outputs from previous sprints and includes packaging and branding. This sprint would not include any new feature. The duration of this sprint may be less than the regular sprint depending on the features added between customer releases.

7. Conclusion and Way Forward

The real life experience in executing Agile in offshore outsourcing with fixed price illustrates the fact that modifications and adaptations (including to the management style and practices) to pure Agile practices can help to overcome the challenges in such scenarios.

However, additional experience data is required to conclude in a definitive way what modifications and adaptations can be applied generically enough to different kind of projects in the organization and also to different organizations.

References

(The links below have been checked to be working fine as on 2nd Feb 2011)

- [1] http://en.wikipedia.org/wiki/Agile_software_development
- [2] [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))
- [3] Manifesto for Agile Software Development; <http://www.agilemanifesto.org/>
- [4] The fifth annual "State of Agile Development" survey, 2010;
http://www.versionone.com/pdf/2010_State_of_Agile_Development_Survey_Results.pdf
- [5] "Agile and CMMI: Better Together"; <http://www.scrumalliance.org/articles/100-agile-and-cmmi-better-together>
- [6] "CMMI or Agile: Why Not Embrace Both!";
<http://www.sei.cmu.edu/library/abstracts/reports/08tn003.cfm>
- [7] "Scrum and CMMI Level 5: The Magic Potion for Code Warriors"; Jeff Sutherland, Carsten R. Jakobsen, Kent Johnson;
http://agile2007.agilealliance.org/downloads/presentations/sutherexp_425.pdf
- [8] "Scrum and CMMI – Going from Good to Great: are you ready-ready to be done-done?"; Carsten Ruseng Jakobsen, Jeff Sutherland;
http://www.agile2009.org/files/WHI0001%20ScrumCMMI%20from%20Good%20to%20Great%201_11.PDF
- [9] "IT majors get more fixed price contracts"; The Hindu Business Line, Monday, Nov 17, 2008;
<http://www.thehindubusinessline.com/2008/11/17/stories/2008111751440200.htm>

About the Authors

Amit Gupta is working as Director Engineering at Hughes Systique Corporation where he is currently managing a large Agile project. He has more than a decade of experience in executing and managing software projects. He has been working with Hughes Systique since last five years and has worked in the past with Aricent. He can be reached at amit.gupta@hsc.com.

Hemanta Chandra Bhatt is currently working with Hughes Systique where he is heading the Quality function. He has more than a decade of experience as quality, process, and excellence professional and has worked in the past with Eaton, Caneum, Keane and Novell. Hemanta has an M. Tech, degree in Quality from the Indian Statistical Institute and B. Tech. in Production Engineering from the G. B. Pant University, Pantnagar. Hemanta is also a certified software quality analyst (CSQA). He can be reached at hemanta.bhatt@hsc.com.

Appendix

(The links below have been checked to be working fine as on 2nd Feb 2011)

SDLC = Software Development Life Cycle

RoI = Return on Investment

Outsourcing = Contracting out of a business function - commonly one previously performed in-house – to an external provider (<http://en.wikipedia.org/wiki/Outsourcing>)

Offshoring = Relocation by a company of a business process from one country to another – typically an operational process, such as manufacturing, or supporting processes, such as accounting (<http://en.wikipedia.org/wiki/Offshoring>)