

# The Agile Offshore Journey

## Best Practices for Agile Offshore Development

# contents

Introduction .....	3
Problem Examination .....	4
Best Practices for Agile Offshore Journey .....	6
Case Study - European Insurance Company .....	11
Conclusion .....	12
Glossary .....	13
References.....	15

## Executive Summary

The popularity of the Agile Development methodology and the acceleration of offshoring coincided during the early 2000's. In its infancy, the Agile methodology did not address use of distributed or remote teams as a large part of its philosophy, instead resting on the premise that teams should be in close proximity. As distributed or remote teams have become the norm, many companies have been working to integrate the Agile methodology with global delivery best practices. **“Go Agile” or “Go Offshore” is no longer a choice an enterprise needs to make.**

Applying the Agile methodology to offshore or the global delivery model has been a challenge. The geographical divide with time zone differences, communication issues due to cultural differences, and the act of marrying time tested traditional development processes with Agile practices are some problems that stand out when contemplating cross-shore Agile projects. Adding to these challenges is how to incorporate Agile processes with the prevailing CMMI processes of offshore centers.

Through L&T Infotech's work with a number of clients, we have discovered realistic ways of mitigating some of the issues with Agile Offshore. Trust based contracts, incrementally adopted Agile practices, carefully choosing the specific Agile methodology and having the fortitude to stick to the chosen method has had positive results in removing the initial challenges in an Agile offshore engagement. Process automation, shared knowledge management and extensive collaboration help during the steady state phase of an Agile project. Last, we realize through experience that excellence in delivery is a direct result of shared expectations. Hence a strong governance and performance mechanism should be established upfront to ensure honest assessments and frequent feedback.

Integrating Agile and CMMI has turned out to be a sweet journey at L&T Infotech. We have largely reduced the risk of Agile project execution offshore by mapping Agile practices to CMMI principles. Our experience reiterates that CMMI should be viewed as a model and not a standard. The goals of CMMI can be achieved by tailoring the process for Agile execution.

With a synthesized approach of Agile and offshore, we are focusing on continually improving the performance of this model to better achieve our client's objectives. This paper highlights L&T Infotech's best practices for running effective Agile engagements in geographically distributed teams.

### Audience

This paper is addressed to:

- Decision makers who need information on offshore possibilities of an Agile project
- Offshore software service providers who are contemplating use of Agile for their development
- Analysts documenting the hurdles and best practices of Agile at offshore

## Introduction

The Agile methodology was formulated to address the shortcomings of traditional development methodologies. For example, the underlying philosophy of traditional, process oriented methods is to have the software requirements completely locked in before the project commences. This is usually impossible to achieve as requirements get refined during the course of development and market conditions may change requiring product features to be reprioritized. In addition, there are multiple levels of abstraction involved in the process of conveying requirements. Traditional methods heavily rely on documentation and hence suffer as much failures as successes in usefulness.<sup>(1)</sup> These inadequacies of rigid requirements and non-living requirement documentation in traditional processes have strongly motivated enterprises to try new solutions such as the Agile methodology that embrace just-in-time requirement specification.

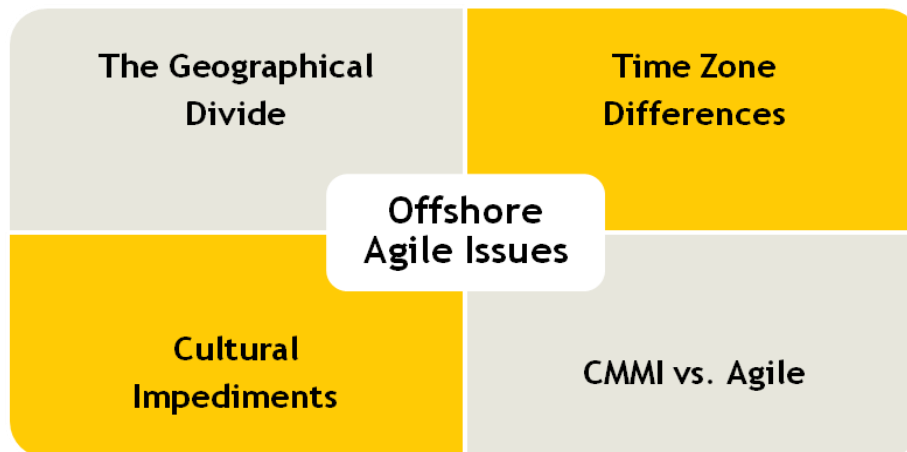
Agile started off as a difficult-to-grasp, odd-ball solution in an arena completely dominated by traditional methods. Today, Agile methods have evoked substantial amount of curiosity, literature and debate among IT professionals. The use of Agile methodology has increased significantly over the past decade. **69% of the small and medium IT development organizations across the globe have one or more Agile projects.** 17% of these organizations have all their development projects using Agile.<sup>(2)</sup> Enterprises have found that utilization of Agile methodology has enabled them to develop products faster with more business alignment than traditional methods.

The popularity of the Agile Development methodology and the acceleration of offshoring interestingly coincided during the early 2000's. In its infancy, the Agile methodology did not address use of distributed or remote teams as a large part of its philosophy, instead resting on the premise that teams should be in close proximity. As distributed or remote teams have become the norm for large companies, many have been working to integrate the Agile methodology with global delivery best practices. **In fact, 47% of the organizations engaged in distributed development have plans to try Agile offshore development in the next year**<sup>(3)</sup>. "Going Agile" or "Going Offshore" is no longer a choice an enterprise needs to make.

This paper will examine some of the issues with implementing Agile in a globally distributed environment and share L&T Infotech's best practices for implementing Agile in an offshore environment.

## Problem Examination

There is lots of skepticism in the market today about the success of Agile offshore development. The reasons center on a few issues, most of which have to do with geography. An in-depth understanding of adaptation problems has allowed us to develop best practices to address these issues on our Agile engagements. The following chart outlines the major problems we see occurring in Agile offshore development.



### The Geographical Divide

Most people experienced in Agile feel that distance forces companies to compromise on the use of Agile. Distributed Agile has unique challenges. However, heightened focus on tests, automation, time boxed iterations and effective usage of collaboration tools bring customers closer and enable teams to deliver right the first time. Delivering initial projects quickly and with high quality, strengthen the user's confidence in the distributed team focusing less attention on distance.

### Time Zone Differences

Globalization currently calls for ways to surmount the time zone obstacle. The process gets tougher with Agile execution unless special efforts are taken to organize the activities of the team. With Agile when code is delivered on a daily basis, prudent activity split based on dependencies between onsite and offshore could convert the time zone difference to be an advantage. For example, builds and test runs can be scheduled to be ready by offshore mornings so that the analysis on any failures is complete and ready to be picked up by onsite mornings. Any productivity loss will be minimized. With mature Agile teams, splitting the project to blocks/modules of independent functionality can help mitigate time zone issues.

### Cultural Impediments

Cultural impediments are generally perceived at the organization level and between the distributed Agile teams. Organizational acceptance for a distributed control style of management is needed for empowerment and trust. Collective ownership, which is a cornerstone of Agile development, suffers as

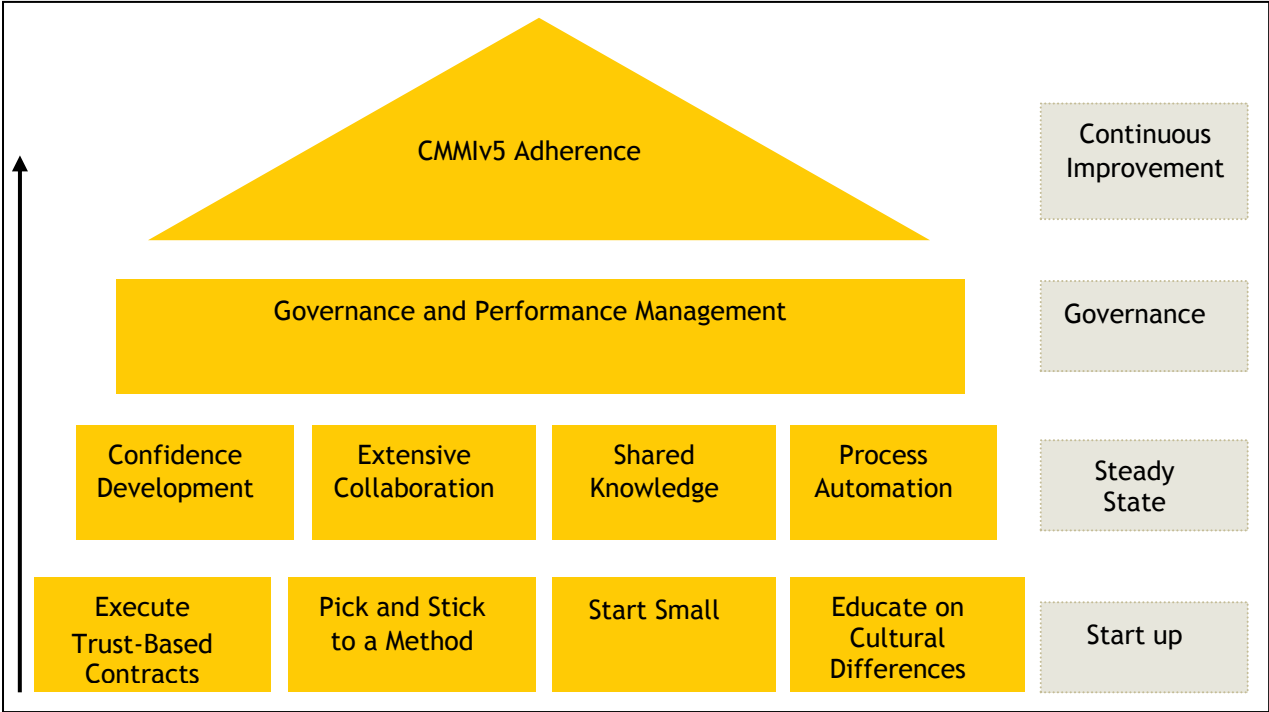
a result of an “*us vs. them*” attitude. The onsite developer, through both formal and informal communications, will have a better understanding of user requirements than an offshore developer. This informal information *or tacit knowledge*, along with the tips gained onshore, need to be passed to the offshore team as the onsite developer uncovers them. This communication needs to be as frequent and as impromptu as possible.

### **CMMI 5 vs. Agile**

Until recently Agile and CMMI were thought to be incompatible. One apparent area of friction seen by the community is empowerment and trust of Agile vs. prescriptive nature of CMMI. CMMI promotes macro level planning at the project level focused on deriving organization level goals, which is not a concern for Agile practices. Offshore service providers have embraced and mastered respectful process models for traditional methods of development such as CMMI. Adapting to the more unconventional development approach like Agile seemed impossible to many. To accommodate Agile development offshore, the prescriptive steps of CMMI should be tailored. CMMI should be viewed as a model and not as a standard. The goals of CMMI can be achieved by tailoring the process for Agile execution. For example the risk mitigation process of CMMI can be tailored for Agile iteration by proposing risk mitigation and tracking within the iteration scope. Any CMMI process tailored for Agile should do so without disturbing Agile principles.

# Best Practices for Agile Offshore Journey

We have found that the layered set of activities below can alleviate issues of Agile offshore and maximize benefits.



## Start-Up Best Practices

### Execute Trust-Based Contracts

Traditional contracts have been drawn without focusing on win-win for all parties concerned. Agile contracts by contrast focus on results for stakeholders. While the sourcing partner views the faster Agile development as a boon to win more contracts, the Client agrees to take on delivery responsibility and that their adequate participation is a critical success factor.

### Pick and Stick to an Agile Method

There are various Agile methodologies available today to cater to the different products under development. Based on the type of the project/product, one should choose a suitable Agile methodology. Crystal methodology could be chosen for mission critical projects. FDD can be used when the development is from the ground up. Scrum and XP can be used if a development framework already exists. Most often, a well defined combination of Scrum management practices and XP engineering practices reap huge benefits.

### Start Small

To start small is not about working from the lower end of the complexity spectrum. It is about allowing the practices to evolve and refine over time. The key part of Agile development is to continuously iterate and improve the software feature by feature. With some meaningful work and an agreement of done definition, a product backlog with user stories can be worked on effectively. Every other improvement can be built based on needs arising out of honest and thorough retrospection. This way the team can choose practices that will help them gain more productivity.

### Educate on Cultural Differences

The cultural differences that become impediments in cross shore projects are more pronounced when the project follows Agile execution. For free flow of information on either side, offshore teams should be groomed with adequate courage to question till they understand. Constant change and gradual refinement is a way of Agile development. Refinements and changes need to be collectively owned. Collective ownership is another learning curve for teams conversant in traditional methods of development. Team spirit could be fostered with team building activities over Video Conference and occasional travel. Developers gradually over time get accustomed to thinking like end-users.

## Steady-State Best Practices

### Confidence Development

Confidence is not a pre-requisite but is developed out of success with each iteration. Slow relocation from onsite to nearshore to offshore has not only imparted technical and domain understanding of the application to be developed but has positively impacted team dynamics and decision making. Agile offshore should start with collocation of the teams during the initial period primarily to improve team dynamics. After initial knowledge transfer of the existing application and code base, a nearshore approach can be adopted and the team can be empowered to make decisions with some oversight. With visible improvements on the capability of the team to handle iterations and decision making, the nearshore team can get transitioned to offshore execution.

### Extensive Collaboration

A unified communication and collaboration model should be adopted by the enterprise. The collaboration model should be simple to use and an alternative should be available. Video conferencing is an effective way for team collaboration. Both parties must ensure adequate availability of the infrastructure needed to support Agile execution. The slower acceptance to tool based collaboration can be overcome by choosing collaboration tools that provide a rich user experience. Our clients use a large variety of collaboration tools such as emails, instant messages, wikis, Webex, webcams, video conferences, digi-pads etc. Some clients also adopt social business software such as Jive, which offers a complete solution for business collaboration.

Availability of a business ambassador at the offshore location can facilitate collaboration to a larger extent especially when developing highly complex functionality. The role of the Business Ambassador is best played by a person with rich domain knowledge who can bring the experience of a business user to the offshore team on a regular basis. A carefully crafted collaboration model that enables face-to-face communication can help flush out assumptions and miscommunications significantly.

### Shared Knowledge Management

Knowledge management should not be segregated between onshore and offshore. Tools such as Wikis, shared directories and defect trackers should be accessible to both the teams over VPN. Calls can be recorded and made available as references for both teams. This is especially useful for meetings that take place in one shore or another.

One of the Agile manifestos, “working software over comprehensive documentation” is often misunderstood as “no documentation”. In Agile it is best to document all the information that might be referenced; a wiki is a popular Agile knowledge management tool that is used for this purpose. Success of wiki is attributed to the freedom it provides. Everything from personal information of the developer, discussions, blogs, help documents, release plans, user guides and patch documentation can be accommodated in a wiki. Small to major process improvements, best practices and lessons learnt are some of the key knowledge areas that will benefit the offshore team’s learning curve.

## Process Automation

Communication, a vital aspect of Agile, is marred by the distance between Agile developers and business users. Over time considerable process automation can be developed that includes acceptance tests. Automated acceptance scripts written by the business user in English-like language helps the offshore team deliver the user story the same way the user intended. Handling complex requirements is less of a problem and can improve over time with experience.

Auto build timings every day can be set to take advantage of the time zone difference. This way, distributed Agile teams can ensure they have not broken the build and caused problems for the other team. Much of the code review process can be left to the analyzer tool integrated to the IDE and to Agile.

## Governance

### Governance and Performance Management

Governance is about consistently achieving established expectations and validating through performance measures. For traditional development the governance and supporting performance measurements are standardized and imbibed into the processes.

For Agile development the metrics to support governance in line with business objectives needs to be defined.

Excellence in delivery is a direct result of shared expectations. For Agile iterations, benchmark criteria can be predefined and can include velocity or scope of iteration, code quality standard, functional and non functional test results, oral & written communication, and teamwork. With the process tailored for Agile execution, metric-based governance is no longer a burden.

Benchmark criteria and retrospective are good tools to use to monitor iteration success. Traditional CMMI based project metrics are useful to measuring the strength of the Agile offshore relation overall.

Distributed Agile teams should measure the Agile maturity of the teams over a quarter to ensure right direction. Practices not yielding value or Agile values requiring improvement can be identified as a result. These lessons are very helpful to scale more Agile projects offshore.

## Continuous Improvement

### Agile CMMI Adherence

CMMI and Agile together present a powerful combination of predictability and adaptability to the marketplace than neither can alone. Many Agile practices are compatible with CMMI. A good understanding of both can help implementing CMMI projects using the Agile methodology, as endorsed by SEI technical paper “CMMI® or Agile: Why Not Embrace Both”<sup>(4)</sup>. The choice or combination of Agile methodologies adopted can determine the CMMI maturity level. CMMI offers the greatest help of de-risking the Agile project. With its continuous process improvement methodology, CMMI prevents degradation of Agile iterations in code and fix cycles. Sprint backlog, reviews and retrospective serve to comply with most of the CMMI Process areas effortlessly. In fact CMMI adherence motivates the team to follow Agile practices with constant fervor. From our experience, it is possible to adhere to CMMI maturity level 5 without the spirit of Agile being compromised.

## Case Study - European Insurance Company

### Background

A leading asset management and life and pension insurance provider in the UK market is a noted early adopter of Agile for product development. With a development framework in place, they had reaped the benefits of Agile, including improved customer satisfaction and more usable products developed with less IT spending. The next step for the client was to have the outsourcing partner be part of the Agile development, though distributed development in Agile had its hiccups in the past.

### Solution

L&T Infotech together with our client adopted a variety of strategies to overcome the challenges of distributed Agile development. In the preparatory stage, we chose the Agile development methodology and held Agile practices familiarization for team resources new to Agile development.

Unlike other Agile methods like FDD that claim individual ownership, collective ownership was preferred, as the application to be developed was a mix of several technologies. A scrum variant was the best fit for this purpose for its quick delivery, collective ownership and project management support.

The lead technical team travelled to the client site to understand the development environment first hand. This was a short phase in which the lead team got acquainted with the different stakeholders of the project improving the team dynamics. This was followed by development from near shore for an iteration to work out the kinks of distributed Agile development. All Agile habits such as task boards and daily huddles along with extensive communication were enabled for the distributed teams with remote collaboration tools. Comfort level with these tools was established and Agile onsite-offshore process was documented to set clear expectations on process control.

The benchmark criteria for team success along with expected milestones were established. The CMMI quality practices were tailored for the Agile project to be executed. To continuously improve throughput and quality of the deliverables, metric goals were defined and tracked. Retrospective findings were shared across the distributed team and immediately incorporated. The wiki knowledge portal was kept updated with decisions and findings. The Agile maturity of the teams was measured over a quarter to ensure we were moving in the right direction.

After 8 weeks (four iterations), our team was transitioned offshore and continued working with one offshore coordinator and one onsite coordinator facilitating the geographical divide. Planning meetings were held with the onsite business users over video conference for all iterations. The entire team was available to each other over online collaboration channels. Mid-iteration show & tell were held from offshore. Within 4 weeks, the offshore team met all the success criteria set for the successful transition.

This was followed by increasing members to the offshore team with a well-defined knowledge transfer (KT) plan. Once they hit the productivity goal established, the offshore team diversified into development on other related applications for the client.

### Results

**We were able to achieve an offshore utilization of 88% for the Agile project.** The deliverables were on time and within budget 93% of the time, residual defect density was 0.04% and code coverage achieved was over 85%. The geographical divide was effectively bridged for Agile

---

communication. The excellence in delivery was assured by the CMMI compliant Agile processes in place.

## Conclusion

Agile offshore is no longer an oxymoron in the industry, although there are very few industry players committed to pursue this challenge. Agile offshoring can be a rich and successful journey for all the stakeholders if the right practices are put into place throughout the journey. Management support and being prepared can help ease initial hiccups common during early stages of Agile offshoring. Ongoing commitment to Agile methodology combined with diligent continuous improvement efforts and incorporation of global delivery best practices such as CMMI can help ensure success of Agile offshoring in the long run.

## Glossary

<b>Agile</b>	<b>Agile</b> is the name for the wider set of ideas that Scrum falls within; the Agile values and principles are captured in the Agile Manifesto
<b>Agile methodologies</b>	<b>Feature Driven Development (FDD)</b> is an iterative and incremental software development process
	<b>Scrum</b> is an iterative incremental framework for managing complex work (such as new product development) commonly used with Agile software development
	<b>Crystal Clear</b> is an Agile methodology for projects with small teams, less than about 10 people in size
	<b>Extreme Programming (XP)</b> is a software engineering methodology which is intended to improve software quality and responsiveness to changing customer requirements
<b>Burndown</b>	See Sprint Burndown
<b>Backlog Item</b>	See Product Backlog Item
<b>CMMI</b>	<b>Capability Maturity Model Integration</b> CMMI is a process improvement approach that provides organizations with the essential elements of effective processes that ultimately improve their performance
<b>Daily Scrum/Huddle</b>	<b>Daily Scrum/Huddle</b> is a fifteen minute daily team meeting to share progress, report impediments and make commitments
<b>Iteration/Sprint</b>	<b>Iteration/Sprint</b> is a short (2-4 week) development cycle focused on delivering an increment of useful business functionality
<b>Done</b>	<b>Done</b> is also referred to as “Done” or “Done Done” this term is used to describe a product increment that is considered releasable; it means that all design
<b>IDE</b>	<b>Integrated Development Environment.</b> IDE is a tool used for source code development
<b>Impediment</b>	<b>Impediment</b> is anything that prevents the team from meeting their potential. If organizational it is the Scrum Master’s responsibility to eliminate it. If it is internal to the team, then they themselves should do away with it
<b>Planning</b>	See Sprint Planning
<b>Product Backlog</b>	<b>Product Backlog</b> is a prioritized list of stories that are waiting to be worked on
<b>Product Backlog Item</b>	<b>Product Backlog Item</b> is a short, time boxed piece of backlog list, which is a user story

<b>Product Owner</b>	<b>Product Owner</b> is a person who holds the vision for the product and is responsible for maintaining, prioritizing and updating the product backlog
<b>Retrospective</b>	<b>Retrospective</b> is a session where the Team and Scrum Master reflect on the process and make commitments to improve
<b>Show &amp; tell</b>	<b>Show &amp; tell</b> is at the end of an iteration or mid iteration where team members share current working functionality to the rest of the team to get feedback and share status
<b>Sprint</b>	Sprint is a time boxed iteration
<b>Sprint Backlog</b>	<b>Sprint Backlog</b> is a plan document containing information about how the team is going to implement the features for the upcoming sprint
<b>Sprint Burndown</b>	<b>Sprint Burndown</b> is a chart that indicates on a daily basis the amount of work remaining in the sprint
<b>Sprint Planning</b>	<b>Sprint Planning</b> is a meeting between the Team and the Product Owner to plan the sprint and arrive at an agreement on the commitment
<b>Sprint Task</b>	<b>Sprint Task</b> is a single small item of work that helps one particular story reach completion
<b>Stakeholder</b>	<b>Stakeholder</b> is anyone external to the team with an interest in the product being developed
<b>Story</b>	<b>Story</b> is a backlog item usually using the template form: as a [user] I want [function] so that [business value] of Product Backlog Item
<b>Scrum Master</b>	<b>Scrum Master</b> is a leader to the team responsible for removing impediments and making sure the process runs smoothly so the team can be as productive as possible
<b>Scrum Meetings</b>	<b>Scrum Meetings</b> are Story Time, Planning, Review
<b>Scrum Roles</b>	<b>Scrum Roles</b> are three: Product owner Scrum Master, Team
<b>Task</b>	See Sprint Task
<b>Taskboard</b>	<b>Taskboard</b> is a wall chart with cards and sticky notes that represent all the work of a team in a given sprint; the task notes are moved across the board to show progress
<b>Team</b>	<b>Team</b> is the development team responsible committing to work
<b>User story</b>	See Story
<b>Velocity</b>	<b>Velocity</b> is the rate at which a team completes work measured in story points
<b>VPN</b>	Virtual private Network
<b>Wiki</b>	User-defined Knowledge portal

## References

1. The Standish Group International, 1995, The CHAOS Report [internet] Massachusetts: Standish Group (published 1995)  
Available at <http://www.standishgroup.com/>
2. Dr.Dobb's Journal 2008 "Has Agile Peaked?"[Online](updated on May 7th 2009)  
Available at <http://www.ambyssoft.com/surveys/AgileFebruary2008.html>  
[Accessed 15 September 2009]
3. VersionOne 2008 Agile development survey 2008 [internet](updated on August 2008)  
Available at <http://www.versionone.com/AgileSurvey2008/>  
[Accessed 10 September 2009]
4. SEI 2008"CMMI® or Agile:Why Not Embrace Both!"[Online](updated November 2008)  
Available at <http://www.sei.cmu.edu/library/abstracts/reports/08tn003.cfm>  
[Accessed 5 September 2009]

## About the Authors

**Devi Abirami V** understands the nuances of different development methodologies and has worked on developing many cutting edge technology applications for L&T Infotech, the latest being Agile distributed development. She has a master degree in computer applications and has a passion to try out different methodologies and evaluate their best practices. She is currently a project lead of software development at L&T Infotech.

**Bharathy G** has a passion for analysis and research on success of Agile offshore methodologies and has over 3 years of experience working on Agile offshore projects. With a master degree in software systems, she works for the Agile process improvement group at L&T Infotech. She is involved in sharing best practices, and supports Agile project start up. She is currently an Agile process improvement group lead with L&T Infotech.

## About L&T Infotech

Larsen & Toubro Infotech Ltd. (L&T Infotech), one of the fastest growing IT Services companies, is ranked 5th globally among the Best IT Services Providers by Global Media Services in 2009, ranked 11th by NASSCOM among the top software and services exporters from India and also ranked among the 'Leaders' category in the prestigious Global 100 list released by the International Association of Outsourcing Professionals (IAOP). A wholly-owned subsidiary of USD 8.5 billion Larsen & Toubro, India's largest technology-driven engineering organization, L&T Infotech is differentiated by the unique Business-to-IT Connect, which is a result of our rich corporate heritage.

We offer comprehensive, end-to-end software solutions and services in the following industry verticals: Banking & Financial Services; Insurance; Energy & Petrochemicals; Manufacturing (Consumer Packaged Goods, High-tech, Industrial Products, Automotive, Chemicals & Process, Media & Entertainment, Pharma, Retail and Logistics); and Product Engineering Services (Telecom).

We also deliver business solutions to our clients in the following Service Lines: SAP, Oracle, Infrastructure Management Services, Testing and Consulting. Our other Service offerings are: Business Analytics, Legacy Modernization, Applications Outsourcing, Architecture Consulting, PLM, Service Oriented Architecture, end-to-end integrated engineering services and embedded system solutions.

For more information, visit us at [www.Lntinfotech.com](http://www.Lntinfotech.com) or email us at [info@Lntinfotech.com](mailto:info@Lntinfotech.com)