



ENHANCING SOFTWARE QUALITY

DISTRIBUTED AGILE

- CHALLENGES & STRATEGIES

PREPARED BY : ANANDA PADMANABAN

TABLE OF CONTENT

1 ABSTRACT.....2

2 AGILE DISTRIBUTED TEAMS - AN INEVITABLE REALITY2

3 COMMON PROBLEMS WITH AGILE DISTRIBUTED TEAMS.....3

4 AGILE DISTRIBUTED TESTING TEAM – STRATEGY3

5 OVERCOME CHALLENGES IN AGILE DISTRIBUTED ENVIRONMENT - BEST PRACTICES5

5.1 COMMUNICATION – ROLE OF TOOLS..... 6

5.2 INTERACTION ACROSS ROLES..... 8

5.3 CAMPFIRE – SHARED SPACE FOR DISTRIBUTED TEAM 8

5.4 OBJECTIVE OF RELEASE – A PREVIEW SESSION 9

5.5 CULTURAL DIFFERENCES – BUILDING TRUST 10

5.6 DISTRIBUTED TEAM MEET & GREET – CREATE A SENSE OF BELONGING 11

5.7 LEAPFROG APPROACH – A MULTI SOLUTION FOR MANY CHALLENGES 11

5.8 OPPORTUNITY TO TRAVEL 12

6 CONCLUSION12

7 REFERENCES13

1 Abstract

Agile software development and the breed of agile methodologies like XP, SCRUM, Feature Driven Development (FDD) and Dynamic System Development Method (DSDM) have gained immense popularity since 2001. The necessity of finding right skilled people, sharing resource and limitation on cost has made distributed software development indispensable. Software development today is more complex than it was 20, 10 or even 5 years ago. Companies routinely perform software development in one or more locations, testing in another location and release and packaging in yet another. This distribution of resources strains the software development infrastructure and raises security, intellectual property and auditing and compliance issues. This paper describes the problems of distributed environment, different strategy and solutions to overcome the problems of agile distributed.

2 Agile Distributed Teams - An Inevitable Reality

In recent years, the vision of IT industry has expanded through distributed team which works across the globe. This becomes inevitable as 57% of agile teams are distributed according to version one survey. Some of the common parameter that companies adopted distributed team is as follows:

Global Talent: Companies go wider their nets, to hire high quality talents across the globe. Visa availability, relocation costs, willingness of the new employee to relocate is the common factors which tend to adopt a distributed team.

Global Market: As businesses expand globally, companies need to gain expertise in those markets. This can be accomplished through mergers or localpartners located in these markets.

Cost-Effective: This is one of the major factors that companies often seek outsourcing the work to region with cheaper development overhead.

Time to Market: Distributed team model helps to handle the pressure of time to market irrespective of time zone differences.

Round the Clock Development: When the level of distribution is wider, software development happens round the clock.

Also, work from home culture and non-availability of resources in the same location are few other reasons which force the companies to go global. Apart from these, there are other factor like advanced communication tools, high bandwidth internet, which influences companies to adopt distribution team. Bug trackers, Source control servers, Web portals and online collaboration tools all help coordinate the distributed projects. Terminal services and virtual machines facilitate remote testing and administration.

Distributed Environment - A bitter truth

We need to accept a fact that team which is distributed is less effective than team which works co-located. In a recent survey of agile projects, the success rate of team located in the same room is 20% higher than geographically distributed teams. Distributed team members always need to put an extra effort to actively communicate and collaborate. Team members need to be flexible on their work timings and schedule in order to continually work.

3 Common Problems with Agile Distributed Teams

- **Exchange of Information:** Root of all problems in the distributed team is communication. One of the agile principles - Face to Face communication - is the substitute for detailed documents. Remote testing team does not generally have an opportunity to engage in Face to Face communication and as a consequence their understanding suffers. Sometimes changes in release plan, requirement changes, defects resolved in the build, projected environment downtimes, etc. are not informed in time to far-located testing teams. The communication channel must be proper to remote teams or else they will be out of context.
- **Knowledge Sharing:** Discussions that require a 10 minute talk in co-located team requires extensive communication and serious management to transform the details to remote team.
- **Working in Different Time Zones:** When the level of distribution gets broader, there would be differences in time zones. When your team works on code/design/testing, it is deep into the night for your far-located workmates. If you struck mid-way of your work there is no way to calling them and getting your work done. You definitely need to wait to get the required clarification.
- **Cultural Differences:** There are different cultural changes. The fact is, people who work in co-located environment have the same kind of cultural approach, but teams which work far-distributed, definitely have difference cultural transformations.

4 Agile Distributed Testing Team – Strategy

- **Build right team over time:** Companies often spend more time to build the right team. Building a right testing team takes time, especially for far-located distributed team rather than near-located team. Identify a Test Lead in order to communicate with distributed development team members, BA, Iteration Manager or Product Manager. The Test Lead from the distributed team

is responsible and the point of contact for all the updates. It is important that team members share a team identity, work towards a common goal and acknowledge the joint responsibility to deliver results.

- An ideal distributed testing team is not a cake-walk for us during initiation of the project. During stand up meetings with distributed development teams, each tester talks about his/her respective defects/issues as rest of the testing team seems to be uninterested in listening to other's issues. Being independent in testing task is good practice but independent in updating status can never be good for a project. Hence we identify an Application Lead who is responsible for the status update, handling calls, clarification and single point co-ordinator for the entire team. Hence there would not be any miscommunication.
- **Build trust between teams:** Group of high profile testers is not called as TEAM. One key factor to build the ideal team for distributed environment is making them trust each other. Create an atmosphere where distributed team members can interact and hence trust will be built over the period of time. Start the day by conducting internal scrum chaired by the Application Lead. Application Lead allocates the cards to the testers, defect to be re-tested & closed and share the key information received from the distributed development team. This way, each tester in the team is aware of the status and is on the same page. Also each team members concerns & issues are addressed.

Application Lead is also responsible for conducting another internal scrum meet for 10 to 15 minutes before jumping into a status call with the distributed development team. By doing so, each one's findings, knowledge, etc. can be shared and communicated.

- **Infrastructure:** It is important to establish and maintain collaboration between the distributed teams and to ensure that the parts developed by teams can be combined into working solutions. Global decisions (e.g. architectural issues, infrastructure or overall testing) should be taking by consensus where possible, to ensure that people are heard and can provide input for the decision and are aware of the decisions that have been taken.
- **Define a control of process:** Miscommunications, misunderstandings and interpersonal conflicts often thrive in the typical distributed team. Companies adopted distributed environment should have process to control the workflow of teams work in multiple locations. It enables continuous delivery by facilitating collaboration between developers, testers and IT operations in large-scale, distributed organizations.

In our distributed testing team each one of the member is trained in all aspects of the test process. All testers in our team well understand the effective of control process in all the areas like defect reporting, test case management and status reporting. The far-located team knows what should be done when the testing environments are down and how much time can be consumed in analyzing a bug. Each process should be documented for the future reference.

- **Use shorter iteration:** More the level of distribution, shorter iteration will work out. It is proven agile teams in distributed environment prefers short iteration say 2 to 3 weeks for successful implementation.
- More the level of distribution, more are the chances of miscommunication. By use of shorter iteration, far-located testing teams are able to cope up with the frequent changes in the product.
- **Maximize the communication channel:** Communication is the key factor for successful product development in distributed environment. Maximize the available communication bandwidth to bring together distributed teams.
- **Choose right tools:** Agile teams cannot rely on sticky notes or burn down chart for the project when distributed. Project artifacts and progress of the project may stay in the same room that needs to be shared across the location. Hence choose the right tools (not just communication) in order to share information among the teams.

5 Overcome Challenges in Agile Distributed Environment - Best Practices

Currently, our team is distributed across the globe where in one of the development team is located at Atlanta-USA, another team is in Canada and the QA team operates from India.

Some of the approaches we have used to overcome the challenges in a distributed agile environment are listed below. These mechanisms have given us fruitful solutions to overcome the problems with distributed agile based working environment.

5.1 Communication – Role of Tools

- **Problem Statement:** One of the major bottlenecks of distributed team is communication. In this, distributed environments communicating to the team which is located on the other side of the globe (India) is more challenging than communicating to the team which is distributed without any time zone difference (Canada).

Since we have adapted to agile methodology, it is certain, for us to adhere to one of its basic principles of “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage”. Some of the high level items are: a) Changes in cards b) Changes in iteration c) Changes in focus d) Changes in release e) Changes in schedule f) Changes in availability g) Changes in priority h) Changes in process. All these changes need to be communicated to the people who are actually involved in it.

In this rapid growth of technology, there are best communication tools available. We need to maximize the utilization of these tools to overcome the communication challenges. Some of them are:

- ✚ **Skype:** There is no substitute for face-to-face communication. We accept it too. But being in distributed teams we are restricted to have physical face-to-face communication. By using Skype video-call, we are able to reduce this barrier to an extent. In the call, we discuss the status of the release, projected defects, plan for the week, etc.
- ✚ **Email:** Email is not the place for documentation, nor is it the place to hold a conversation. Email is for broadband announcements, overall status updates on systems, questions for the whole company and other bits of general information. We take fullest advantage of Email to share the information across the team. Meeting invites, end of day status report, clarification required, personal take-off reminder (PTO), follow ups are all shared through Email.
- ✚ **IM:** One of the easiest ways to access team members to share quick information / confirmation is through instant messaging. We use AOL to have one-to-one and group chats. This is an informal way of exchanging information before we publish it formally through Emails.

- ✚ **VOIP:** A voice over IP system is important for daily stand-ups and for conversations. VOIP is scalable and provides interoperability. Information we require is just a phone call away.

- ✚ **Web-ex:** Resources keep changing due to various reasons like swapping between projects or leaving the organization. Training the new resource is vital for any project. Web-ex is a highly interactive tool which is used to have training or knowledge transfer sessions.

- ✚ **Web-Conference:** A must have tool in every design or architectural meetings.

- ✚ **WIKI:** Highly customizable. The wiki is where the team keeps all of its institutional and historical knowledge. It is the single point of reference for documentation on the system and the team's process.

- ✚ **Desktop Sharing:** GoToMeeting, provides the ability for presenters to share their desktop. This is where we review our burn down charts, etc. Another instance would be, when a defect is not re-producible at the developer's end, we can send an invite to view our desktop.

	Daily Status	Video Call	Clarifications	Unplanned Calls	Bug Reproduce	Cards Preview Session	Knowledge Sharing Sessi	Cards Estimatio	Project Common Information
Skype	●	●						●	
IM			●						
EMAIL	●		●						
VOIP	●			●				●	
WIKI									●
Web Ex						●	●		
Go To Meeting					●				

Figure (i) – Communication Layers: Some of the tools in our current project

Due to usage of communication tools effectively we have reduced the % of dropped out cards in the past few releases.

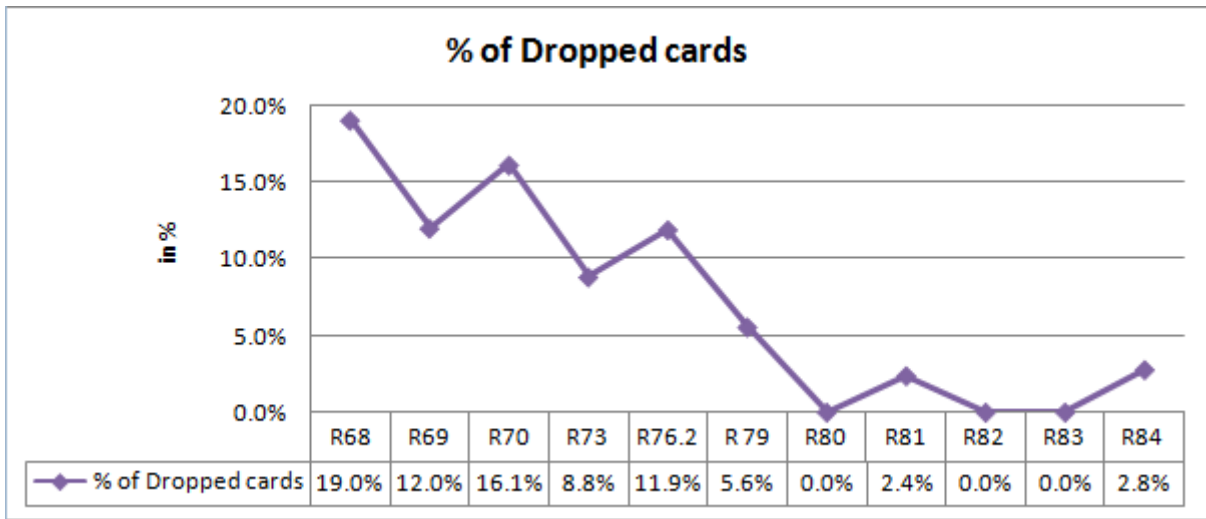


Figure (ii) – Example of % of Dropped Cards/Requirements in a project

5.2 Interaction Across Roles

Getting right information makes sense only if it is received at the right time. For an instance, our QA team which works in India, looking for some information from a developer who works in US. Instead of QA team sending out an email to QA Lead who works in US, we directly post the question to the developer by keeping others in the loop. Similarly, if a developer who works in US needs a clarification from a BA in Canada; we let the developer directly post the question to the BA by copying others who ought to know.

5.3 Campfire – Shared Space for Distributed Team

Campfire is a web-based group chat tool that lets you set up password-protected chat rooms in just seconds. Invite a client, colleague or vendor to chat, collaborate and make decisions. Currently our team is distributed with a huge time zone difference. There are several discussions that happen every day during onshore time. Just to bring all the distributed teams in loop, we identified a dedicated machine which we never turn off and we login into campfire when we leave the office. The next morning, we can read the previous day's discussion and that way we are updated. If we have any questions on the discussion, we can post or mail the concerned person. This is another way to share information discussed in a distributed team.

Direct interaction across the roles and campfire helped us in reducing the QA environment down time drastically. The below listed figure depicts the number of QA downtime hours reduced in the past few releases.

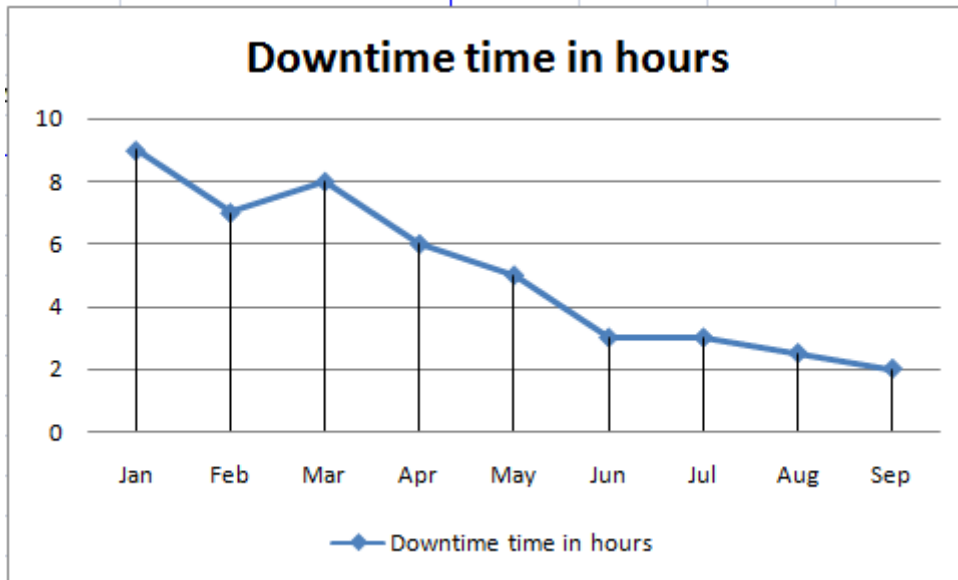


Figure (iii) – The number of QA downtime hours reduced in the past few releases

5.4 Objective of Release – A Preview Session

Team members may be geographically distributed but not the goal of the project. Everyone in the team should have a common objective. During initial phase of the release, we conduct a kick-off meeting, chaired by the Iteration Manager & BA in order to explain the objective of the release like total agreed points for delivery, how cards are related to each other and severity of the release. The preview session contains some mock-up, demo in order to understand the objective of the release. As a result we have reduced the number of as designed (Invalid) defects for the past few releases.

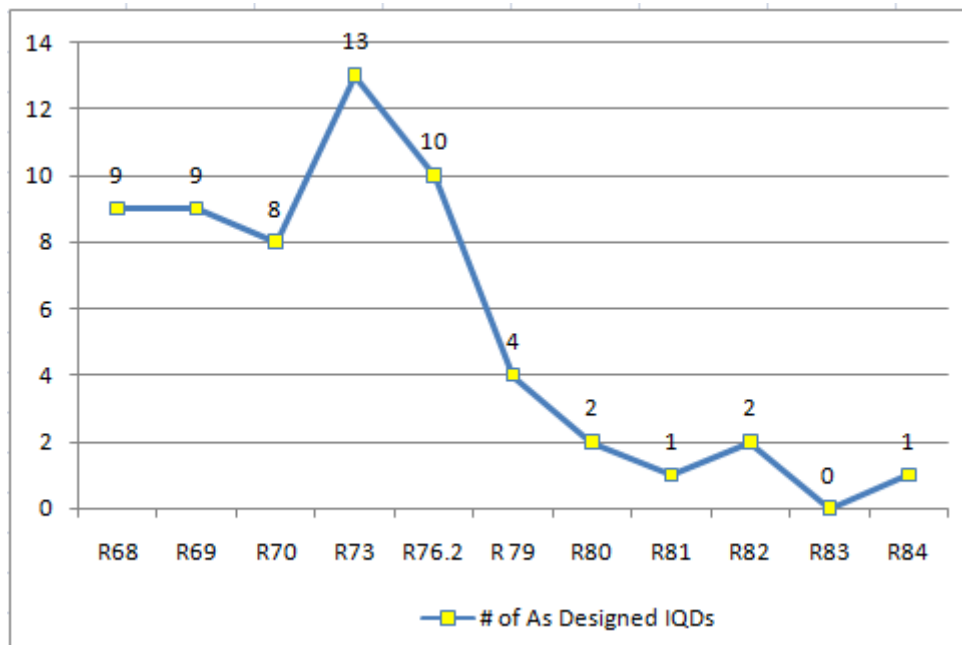


Figure (iv) – The number of invalid defects in the past few releases

5.5 Cultural Differences – Building Trust

Team across the globe definitely has cultural issues to deal with. What is cultural difference? Experts see this as two - Organizational & National.

🚩 **Organization Cultural:** “Order & Obey” model still active across the organization. Managers make the decisions and low level team members acknowledge it. If your onshore team makes decision, offshore team exists just to implement the tasks. This prevents people from asking questions, talking about deadlines, proposing alternative solutions, etc. Some of the resolutions for the above problem are to develop openness, visibility, by bringing in the culture to encourage any deserving team, trust them and give responsibility to who actually does the work.

🚩 **National Cultural Changes:** (I apologize if the content under this title sounds offending; it is not meant to be personal.) When topics regarding ‘nation/country’ pop-up, people get emotional. Some people sound authoritative whereas others talk politely. It is just a simple culture change. Saying “No” to something is different from saying “I guess this may not work out, please review it yourself once again”. Create an atmosphere where each person

feels safe to raise issues and ask questions. Yes, this is something you should do on co-located teams too and it is even more critical when time and distance separates a software development team.

5.6 Distributed Team Meet & Greet – Create a Sense of Belonging

This is a fun session between distributed team members to get to know each other. Though it is not related to project activities, by this approach we motivate our team members to interact with each other better. All team members participate in this meet. Each team demonstrates their skill by presenting on any topic. And the novel presentation is awarded. This gives them the sense of belonging. This indirectly helps in the project to achieve the common goal.

5.7 Leapfrog Approach – A Multi Solution for Many Challenges

When working in a distributed environment there always arises a problem of handling work effectively; Miscommunication persists. Also when the team is geographically working at different time zones, with different holiday calendars there exists a problem of taking the leadership in presenting the right kind of work and delivery.

Using the leap frog approach we are able to work independently as every single task is communicated to both the onshore and the offshore team (either from onshore or offshore). We need to trust our team members and keep in mind that all of them are skilled and efficient to work on their tasks. Here we try to apportion the work to all of them, i.e. developers, testers, etc. irrespective of the location and give the opportunity for each team to lead a release. So by doing this, every team knows what needs to be. This makes them more responsible and helps to enhance their work culture also. This increases the trust on the team and creates a positive vibe among the team.

Due to adoption of leap frog approach and by minimizing the cultural differences the quality of our QA grade has been steadily improving in the past few releases. (QA grade is the assessment of QA team work at the end of each release.)

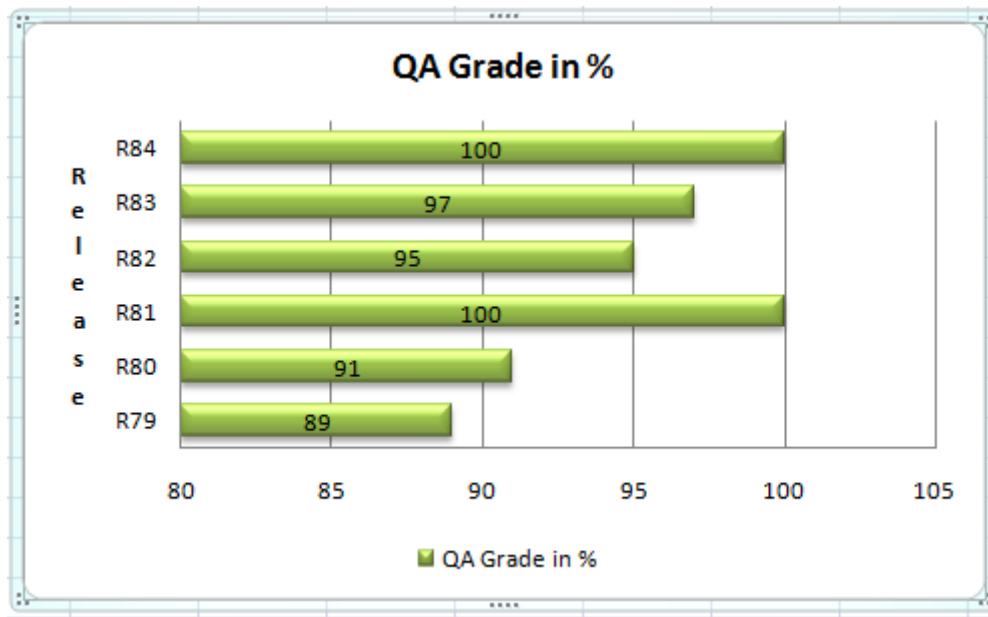


Figure (v) – QA downtime hours reduced in the past few releases

5.8 Opportunity to Travel

Plan a travel for deserving candidates as a rotation to work closely within a distributed team. We plan it for short durations of a minimum of 3 weeks. Everyone would love to venture overseas and handful of overseas allowance. Who would say “No”? Since this is meant only for deserving candidates, it motivates everyone in the team to prove they are productive. This approach not only enables the team member to understand the culture but also the working approach of a distributed team.

6 Conclusion

Distributed agile needs a little adaption to the agile practices in order to be successful. Communication and collaboration are the key factors to work on distributed environments because software development is 80% communication and 20% everything else. Understanding of the culture would create a stronger team bond. Being willing to invest in travel, doing a bit more up front modelling, organizing your team around the architecture and adopting more sophisticated tools are the key to your success with distributed agile. Though it has some challenges, it is up to the people who take it up and implement it successfully, thereby leveraging rapid business value delivery.

7 References

1. "Agile Software Engineering", Orit Hazzan, Yael Dubinsky, Springer Verlag-London limited, 2010
2. "A Practical Guide to Distributed Scrum ", Elizabeth Woodward, Steffan Surdek, Matthew Gani, 2010
3. "Agile Product Management with Scrum: Creating Products that Customers Love", Roman Pichle, 2010

About Indium:

Indium Software is exclusively focused independent software testing services firm since 1999. Over the years, Indium mastered objective methods that minimize the risk of failure of applications and software products. With a global headcount of over 300 employees, Indium works for a mix of marquee Enterprise and ISV clients spread across the globe. Indium is aggressively pursuing the Social, Mobile & Cloud agenda to make these the core of our next wave of service specialization.

Contact us:

USA - Indium Software Inc.

Corporate Office California

Suite 210, 1250 Oakmead Parkway
Sunnyvale, CA - 94085.
Phone: +1(408) 501-8844
Fax: +1(408) 501-8808

Atlanta

Crown Office Suites,
1870 The Exchange, Suite 100,
Atlanta, Georgia 30339. USA
Phone: +1(770) 989-7302,
+1(678) 608-0809

Sales Inquiries

americas.sales@indiumsoft.com
apac.sales@indiumsoft.com
emea.sales@indiumsoft.com
india.sales@indiumsoft.com
sales@indiumsoft.com

INDIA - Indium Software (India) Limited

Global Delivery Headquarters Chennai

VDS House
41, 2nd Floor
Cathedral Road
Chennai - 600 086.
Phone: +91-44-2811 6330
Fax: +91-44-4210 4033

Prince Arcade Building
22-A, 3rd floor
Cathedral Road
Chennai – 600 086.
Phone: +91-44-4347 7200

Steeple Reach,
Old No. 25, New No. 39,
Cathedral Road, Chennai –
600 086.
Phone: +91-44-4347 7122

General Enquiries

careers@indiumsoft.com
info@indiumsoft.com

Bengaluru

Salarpuria Adonis, 2nd Floor
Binnamangala, 3/1, Old Madras Road
Kadranapalya, Indira Nagar
Bengaluru – 560 038.
Phone: +91-80-6784 7500

New Delhi

F-1/5 Okhla Industrial Area,
Phase-I,
New Delhi – 110 020.
Phone: +91-11-6613 0400