agile faqs

Organizational Wide Agile Adoption Case Study

Naresh Jain
naresh@agilefaqs.com

agile faqs

Agenda

- ☑ Background
- ☑ Issues teams were facing
- ☑ Readiness Assessment
  - ☑ Results
- ☑ Agile Adoption Road Map
- ☑ Results

agile faqs

Background

- Off shore Development Centre (ODC) for a large Automobile manufacturer
  - Team size 350+
    - Consisting of developers, testers, BAs, DBAs and PM
- Nature of Work
  - New Development
  - Maintenance and Enhancement
  - Staff Augmentation
- Already started implementing Scrum on some of their teams
- Company was a CMM Level 4 company and had its own version evolved from RUP.

Issues they were facing

- There were communication gaps, which lead to lot of rework
- Not in a position to adapt to the customer's & market's needs quickly
- Quality was getting compromised
- On-site team wanted more collaboration
  - There was lot of micro-management from onsite, but still they did not feel like they had the situation under control
- High Attrition rate in India
- Overall frustration on both ends

agile faqs

How we got involved

- ☑ Onsite Management was sold on Agile and they wanted to introduce Agile
- ☑ They got some training and consulting onsite
- ☑ They had mixed results on trying Agile in India
- ☑ They realized that to take full advantage of Agile, they would need an expert team to help them with the adoption.
- ☑ Onsite team contacted us to help with create a road-map for their organization wide transition to Agile (Scrum + XP)

We Started off with

- Week long Readiness Assessment
  - Before attempting to make an organizational change, it's prudent to learn whether you're planting your seeds in soil that will allow for future growth. Some organizations are not fertile ground for accepting new ideas, such as XP or technical practices like Test-Driven Development. It is critical to discover whether team members are interested in change and willing to try something new as well as whether the organization allows for change to occur.
- Goals of this Readiness Assessments were:
  - At a very high-level, understand various process used throughout the Software Development Life Cycle (SDLC)
  - To study the current understanding and implementation of Agile/Scrum practices
  - To identify a pilot project, which will be used to coach the team about Scrum and eXtreme Programming
  - Plan Agile/Scrum/XP training based on the preliminary gaps identified.
  - Identify Agile adoption strategy for Distributed Projects
  - Identify some local champions who can potentially become local Agile Coaches.

Things Identified During the Assessment

Issues arising because of
Distributed nature of development
&
Current Agile/Scrum/XP knowledge levels of teams @ both locations

Distributed Dev: Needs Attention

- ☑ General
- ☑ Management
- ☑ Communication
- ☑ Tools and Environment

General

- Good support to try Scrum, but teams lacked clarity about its applicability and suitability for offshore projects and had open questions about the same.
- Organizational (Offshore) Cultural Difference
  - Its common for team members to rotate projects every 18-24 months
  - Team members didn't know how to say "No".
- Some team members on projects lacked business/domain knowledge.
- Ramp-up time for new team members took a long time and caused frustration on both side

Management

- Some teams were very closely monitored by onsite team member. Sometimes this felt micro-management-ish. Teams thought Scrum was the culprit.
- Both sides lacked trust.
  - Team members hesitated taking ownership of the project because they didn't see the big picture
- Scope was not reduced by the Product Owner when new items were added to the current sprint. This was causing burn-out
- In the past, there were few last minute changes done during the sprint.
  - Ex: UI Guidelines and DB Changes

Communication

- Studying some of the bug reports, We noticed a lot of back and forth communication during the bug analysis phase.
- Communication leaks - some business rules or functionality drops off during development because they are not documented along with requirements as acceptance criteria
- Most team members didn't have or could not see the big picture. They were just asked to focus on the next sprint's work in the name of agility. This made them team feel like there was no proper planning and requirements were very vague. The team felt there is no real time during planning to think through all the use case scenarios. Overview of the overall functionality of the application would have been helpful.

agile faqs

Tools and Environment

- Development on Virtual PCs was slow
- Major differences in infrastructure and data existed in production, testing and dev environment. In some cases they were not even 50% close.
- When the application integrated with other applications, end-to-end integration testing was not possible offshore. This leads to bugs that could only be produced onsite.
- Teams didn't have read-only access to production database. This made it difficult to solve production bugs.
  - Due to security reasons the read-only access had been removed. Now the team has to send the query to someone onsite and they execute the query and sent the results back.

agile faqs

Agile Knowledge Level

- ☑ Needs Attention
- ☑ General
- ☑ Process
- ☑ Code Quality
- ☑ Testing

General

- Lots of myths about Agile
  - Concerns around the low amount of documentation
- Scrum/Agile was not well understood by both onsite & offshore teams and in some cases it was creating the wrong impression.
  - Scrum was being forced upon the teams as a silver bullet, without everyone's buy-in
  - Scrum was giving onsite team tight control over everything. Leading to micro-management
  - Lacked big-picture understanding: Only bits and pieces of Scrum was implemented by the team
- Different teams were at different levels of Agile adoption. There were quite a few differences in their approach
  - Some had a distributed standup meeting while others just had a local standup meeting
  - Some used comprehensive use cases and activity diagrams while others just used light-weight documents

Process

- In very early stages of implementing Agile. Most teams followed a mix of traditional iterative process and Agile. But most teams were still quite waterfallish.
  - People were sitting in the mould where they wanted
    - Complete requirements specified upfront
    - Proper detailed Designs - At least in words/flows if not in sequence/activity diagrams
    - No major changes throughout the project
    - Zero regression bugs without any automated test safety.
  - Teams were not really self-organizing
    - Teams were not comfortable with any team member pulling tasks from the board. This was a challenge since not everyone was comfortable and knowledgeable of all the technologies used on the project
  - Teams were not structured (cross functional) to work in Scrum model
    - Tester to Developer ratio was quite low (1:8)
    - Offshore teams lacked (proxy) Product Owner and a person to make Architectural decisions

Process...

- Huge amount of manual work. Manual Testing, manual build & deployment process and manual DB updation process.
- Most teams didn't have a Daily Scrum Meetings. On teams which had them, they were turning into long status meetings
- Teams went through phases of high pressure and relaxation (lack of sustainable pace)
- Most teams relied heavily on heavy weight documentation.
  - Big upfront analysis and design process.
  - Very very Detailed Activity Diagrams were created by on-site team members. Developers didn't really have to think much.

Process...

- Except on one team, most teams used traditional methods for Estimation and Tracking
  - PM estimated the functionality. Sometimes took help from module leads
  - Gantt charts for tracking
- At times there were lots of support tickets and only one person addressing them. This lead to a large backlog for production issues.
- User requirements were not being Thin sliced
  - Tasks were mostly divided along the architectural layer
- On Maintenance projects the iterations were 3-6 months long. Almost 6 weeks was spent on planning and estimation.
  - Teams were providing detailed estimates for bugs which were very tricky to estimate

Code Quality

- Based on some quick sampling of code on some projects, we noticed:
  - No automated Unit tests
  - Many layers in the code, all of them just delegating to the next layer.
  - Code duplication in several places. Based on what we saw, we think this code base would be difficult to maintain.
  - The Databases in various environments like local, QA and production were not in sync. This was causing difference in code's behavior.

Testing

- There were bugs which could only be reproduced in the QA environment. Local and System Testing environment could not reproduce those bugs. This lead to depending on onsite QA Test for some bugs
- Scenario based regression testing was done manually. Even selected regression tests were taking a lot of time at the end of each sprint. Because of which
  - Code was frozen 2 weeks in advance and hence massively impacted the productivity of the team.
  - Also this style of regression testing was not very effective since not all testers knew the overall system and the over-all impact of certain functionality.
- On some projects quite a few look & feel (GUI) related bugs were re-surfacing all the time

What was missing on most teams

- ☑ Clear understanding about Agile/Scrum and the rationale behind it
- ☑ Agile Evangelist and internal coaches who could facilitate knowledge sharing between and within various teams implementing Agile
- ☑ Incremental release and Small iterations (fixed time boxes)
- ☑ Release and Sprint planning involving the whole team
- ☑ User stories to represent project requirements
- ☑ Cross functional Teams in each location (at least proxy Product Owner)
  - ☑ User Experience and Usability skills
  - ☑ Decentralized decision making powers
  - ☑ Cross team collaboration (Scrum of Scrum model) with local team representatives

What is missing on most teams...

- ☑ Cross Pollination & Ambassadors (both locations)
- ☑ Code
  - ☑ Evolutionary Design
  - ☑ Good Coding practices based on Object Oriented Design Principles and Patterns
  - ☑ Test Driven Development and Refactoring
- ☑ Testing
  - ☑ Automated Acceptance tests
  - ☑ Automated UI Tests
  - ☑ Performance (Load and Stress) Testing

agile faqs

What is missing on most teams...

- ☑ Automated Build and Deployment process
  - ☑ Static Analysis tool
  - ☑ Automated build and Continuous Integration
  - ☑ Test Coverage
- ☑ Informative workspaces and Information Radiators
  - ☑ Big visible charts
  - ☑ Story wall, etc
  - ☑ Wikis
- ☑ War room style team rooms

agile faqs

What is missing on most teams...

- ☑ Metric to measure true quality and project status
- ☑ Retrospectives & Real Standup meetings
- ☑ Sustainable pace (Team members are getting burnt out)
- ☑ Proper tool to handle the product and sprint backlog
- ☑ Rotation plans for the team members

agile faqs

Post Assessments: Next Steps

- ☑ Identify a Pilot Project
- ☑ Identify Internal Agile Coaches
- ☑ Increase Organization wide Agile awareness through Trainings and Interest Group based discussion sessions
- ☑ Establish a Distributed Agile Adoption Strategy

agile faqs

Pilot Project Selection Criteria

- ☑ A critical project (high-visibility) where team members were open to accepting change
- ☑ Project which was not in a state of beyond-recovery
- ☑ Decent sized project where we could see the real impact of introducing Agile

agile faqs

Internal Agile Coaches Identification

- Identified One person per team
  - Motivated and passionate!
  - In leadership role with strong technical skills
  - Good understanding/handle on how the team works (its process)
  - Individual who have earned respect from rest of the team (Go-to person)
  - Who will be co-located with the team (offshore) for the next 6-12 months
- Ideally such individuals are under high demand from customers
  - They will need some bandwidth to perform these tasks
    - For now they need to spend a day a week for the next 1 months learning about Agile
      - Self-learning, classroom training, on-job
- We also ensured we had a good diversity

agile faqs

Organization wide Agile Training

- ☑ 2 day Basic Overview Training for all team members
- ☑ 3 day Project Management Workshop for Team Leads and PM
- ☑ 2 Day Project Automation Workshop for selected team members
- ☑ 2 Day User Stories Workshop for BAs and Tech Leads
- ☑ 3 Day TDD and Refactoring Workshop for all Developers, Architects & TL
- ☑ 2 Day Agile Testing Workshop for all Testers & Tech Leads
- ☑ On going coaching and mentoring of identified coaches

agile faqs

Agile Adoption Strategy

- Before we started pushing Scrum on teams, we had to
  - Do a basic overview training.
    - Help the teams understand the rationale (get out of the waterfall mindset)
  - Address their concerns
  - Explain the adoption strategy, get their buy-in (avoid forcing it down on them)
  - Get enough momentum before flipping the switch
- When a new project starts up, have 2-3 development sprints onsite before starting offshore development
- Identify a pilot project to create a success story

agile faqs

Road Map 1

* Assuming we'll be available one week a month

agile faqs

Road Map 1

* Assuming we'll be available one week a month

agile faqs

Road Map 1

Overview Training
Identify Pilot Project
Publish initial Report

* Assuming we'll be available one week a month

agile faqs

Road Map 1

Hands-on Training
- Project Automation
- Identifying and Writing Effective User Stories
- TDD + Refactoring + Mocking + Legacy Code
- Testing Workshop
- Scrum Process (Standup, Retrospective, Review, Planning)

Identify Internal Coaches
Identify Scrum Master on Pilot Project
Document current state of Project
Define Metric to Measure
Identify potential automation and testing tools to use

* Assuming we'll be available one week a month

agile faqs

Road Map 1

Review progress on Pilot Project
Coach Pilot Project on
- Project Automation
- Pair with BA (User Stories + Acceptance Tests)
- Pair with Devs (TDD and Refactoring)
- Pair with Testers (Automated Tests)
- Pair with ScrumMaster to facilitate planning, review, retrospective, standup meeting, storywall, metric, etc

Summarize each day's progress to other internal coaches

* Assuming we'll be available one week a month

agile faqs

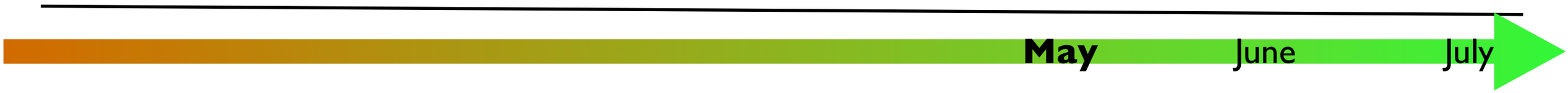
Road Map 1

Continue Coaching Pilot Project
Continue Mentoring Internal Coaches

* Assuming we'll be available one week a month

agile faqs

Road Map 1

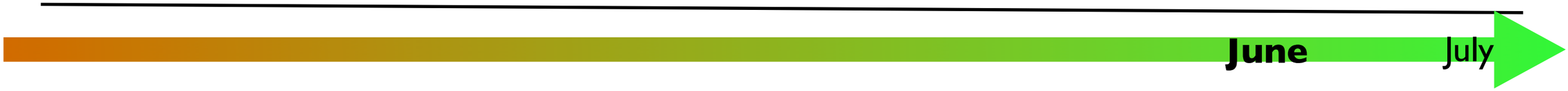


Continue Coaching Pilot Project
Continue Mentoring Internal Coaches
Identify the next pilot project

* Assuming we'll be available one week a month

agile faqs

Road Map 1

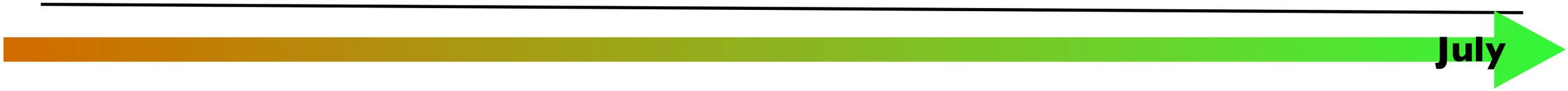


Coach new Pilot Project
Continue Mentoring
Internal Coaches

* Assuming we'll be available one week a month

agile faqs

Road Map 1



Continue Coaching
Pilot Project
Continue Mentoring
Internal Coaches

* Assuming we'll be available one week a month

Road Map 11

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

Overview Training
Identify Pilot Project
Publish initial Report

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

Visit 1
Hands-on Training
- Project Automation
- Identifying and Writing Effective User Stories
- TDD + Refactoring + Mocking + Legacy Code
- Testing Workshop
- Scrum Process (Standup, Retrospective, Review, Planning)

Identify Internal Coaches
Identify Scrum Master on Pilot Project
Document current state of Project
Define Metric to Measure
Identify potential automation and testing tools to use

Visit 2
Review progress on Pilot Project
Coach Pilot Project on
- Project Automation
- Pair with BA (User Stories + Acceptance Tests)
- Pair with Devs (TDD and Refactoring)
- Pair with Testers (Automated Tests)
- Pair with ScrumMaster to facilitate planning, review, retrospective, standup meeting, storywall, metric, etc

Summarize each day's progress to other internal coaches

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

Continue Coaching Pilot Project
Continue Mentoring Internal Coaches
Identify the next pilot project

* Assuming I'll be available two weeks a month

agile faqs

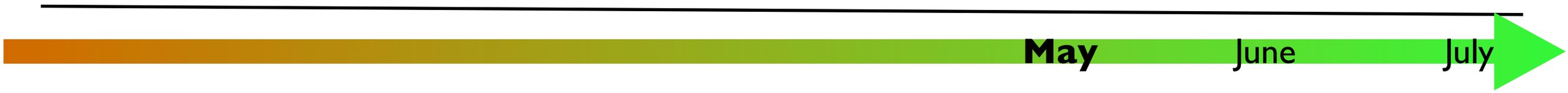
Road Map 11

Coaching new Pilot Project
Continue Mentoring Internal Coaches
Do selected training for other teams

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

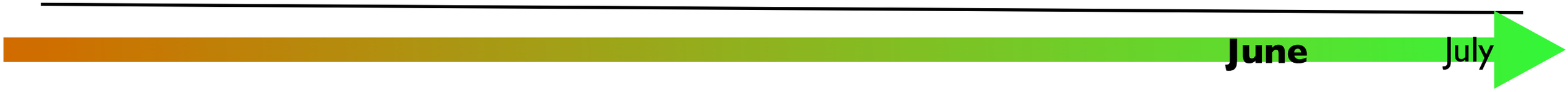


Continue Coaching Pilot Project
Continue Mentoring Internal Coaches
Identify the next pilot project

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11

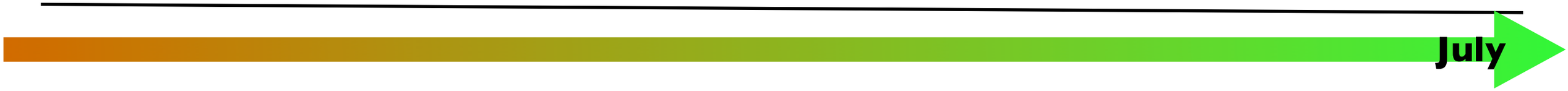


Coach new Pilot Project
Continue Mentoring
  Internal Coaches
Do selected training for
  other teams

* Assuming I'll be available two weeks a month

agile faqs

Road Map 11



Continue Coaching
Pilot Project
Continue Mentoring
Internal Coaches

* Assuming I'll be available two weeks a month

After 18 Months...

- ☑ We choose Road Map II
- ☑ 13 Teams have fully embraced Agile (Scrum & XP) and are self-sufficient.
- ☑ We have 16 Internal Coaches who are helping other teams
- ☑ Customer satisfaction gone up from 2.5 to 3.75 (on a scale of 1-5)
- ☑ On an average turn-around time for
  - ☑ New features gone down from 4 months to 1 month
  - ☑ Bugs gone down from 3 weeks to 1 week
- ☑ Attrition rate dropped from 52% to 14%.
- ☑ All team members have consistently got a rating of 4.5 in their performance appraisal

Thank you!

naresh@agilefaqs.com