This report contains...

- Expectations of our visit
- Summary of our visit
- Our current understanding of the overall team & process
- Results from an Agile Adoption Retrospective conducted with the team
- Observations: What we found
- Conclusion and Recommendations
- Plan for the next 6 months
Expectations of our visit
Objective

Formulate a strategy (plan) to improve team’s effectiveness and to take the teams to the next level
We had to understand...

- Current state of affairs and how the teams operate
- Key challenges faced by the team and its management
- Key stakeholders and their expectations
- Gaps in terms of knowledge and skill levels of the team
- Historic data to understand the performance of the team
Deliverables of this Assessment

- Agile Adoption Roadmap (with specific recommendations for training/coaching)
- Identify Potential Internal Coaches
- Risks Backlog
- Agile training
Summary of our visit
What was done

- Discussion with the teams to understand how they operate as of today
  - Process, Team composition, etc.
  - Current Org. structure
- Value Stream mapping of the overall process
  - To identify bottlenecks and gaps
- Retrospective
  - To understand current pain points
- Watching standup meetings
  - Observe team’s behavior
What was done...

- Rationale Behind Agile (Point Ball Game) Training

- Meeting with important stakeholders to understand their expectations and pain points

- Pairing with Developers
  - Technical Debt, Code Quality, CI and other infrastructure

- Pairing with the Testers
  - Building Quality into the Process

- Discussions with Scrum Masters
  - Planning, Estimation, Velocity, Retrospectives, etc.

- Discussions with the Product Owners
  - User Stories, Product Backlog, Acceptance Criteria, Reviews
<table>
<thead>
<tr>
<th>Activity</th>
<th>Who</th>
<th>Duration</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Process Understanding</td>
<td>Various Team Members</td>
<td>10 hr</td>
<td>Mon-Friday</td>
</tr>
<tr>
<td>Value Stream Map</td>
<td>POs and SMs From Each Team</td>
<td>5 hrs</td>
<td>Mon-Friday</td>
</tr>
<tr>
<td>Overall Agile Adoption Check</td>
<td>Whole Team</td>
<td>2 hrs</td>
<td>Monday</td>
</tr>
<tr>
<td>Meeting with different stakeholders</td>
<td>Stakeholders</td>
<td>1.5 hrs</td>
<td>Thur</td>
</tr>
<tr>
<td>Pair with Developer</td>
<td>Devs from different Teams</td>
<td>2 hrs with each developer</td>
<td>Mon-Thursday</td>
</tr>
<tr>
<td>Pair with Tester</td>
<td>Tests from different Teams</td>
<td>2 hrs with each tester</td>
<td>Mon-Thursday</td>
</tr>
<tr>
<td>Pair with PO</td>
<td>POs from most Teams</td>
<td>2 hrs with each PO</td>
<td>Mon-Friday</td>
</tr>
<tr>
<td>Pair with SMs</td>
<td>SMs from few Teams</td>
<td>2 hrs with each SM</td>
<td>Mon-Friday</td>
</tr>
<tr>
<td>Rationale Behind Agile</td>
<td>Whole Team</td>
<td>2 hrs</td>
<td>Tue</td>
</tr>
<tr>
<td>Open Q&amp;A</td>
<td>Whole Team</td>
<td>1 hr</td>
<td>Friday</td>
</tr>
<tr>
<td>Roadmap</td>
<td>Potential Coach</td>
<td>4 hrs</td>
<td>Friday</td>
</tr>
</tbody>
</table>
Current Process

Our Understanding So Far
Team Structure with roles and responsibilities
Overall Process Flow

For every request:

1. Pre-Planning Meeting
2. PSI
3. Sprint
4. PO + SM + Scrum Team

Product Backlog:
- PLMT
- Customer Team
  - Product Manager
  - Marketing Rep
  - Customer Support Rep
- PSM
- Epic
- Rally
- PO

PSI:
- PSI 1
- PSI 2
- PSI 3

Release:
- Release 1: 2 months
- Release 2: 2 months
- Release 3: 2 months

Sprint:
- Sprint 1
- Sprint 2
- Sprint 3
- Sprint 4

2 months

2 Weeks

Copyright © 2005-14, AgileFAQs. All Rights Reserved.
Customer Requirement Flow

Customer Team Backlog
- Epic
- Features
- Epic
- Product Line Management Team
- High Level Stories
- High Level Stories
- Defects
- NCCA
- Rally

Product Backlog
- PO + Uber PO + Customer Team
- PO + SM + All Scrum Teams
- PSI Plan
- Risk & Impediments
- Dependencies
- Story Point Estimates

Sprint Backlog
- Low Level Stories
- Low Level Stories
- Defects
- NCCA
- Rally

Copyright © 2005-14, AgileFAQs. All Rights Reserved.
<table>
<thead>
<tr>
<th></th>
<th>Wed</th>
<th></th>
<th>Tue</th>
<th></th>
<th>Tue</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S1</strong></td>
<td></td>
<td>Retro</td>
<td>1 Meeting</td>
<td>PO</td>
<td>Sprint Demo</td>
<td>1-1.5 hrs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Planning</td>
<td>1-1.5 hrs</td>
<td>PO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
<td>1 Meeting with Customer Team</td>
<td>PO</td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
</tr>
<tr>
<td><strong>S2</strong></td>
<td></td>
<td>Retro</td>
<td>1 Meeting</td>
<td>PO</td>
<td>Sprint Demo</td>
<td>1-1.5 hrs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Planning</td>
<td>1-1.5 hrs</td>
<td>PO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
<td>1 Meeting with Customer Team</td>
<td>PO</td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
</tr>
<tr>
<td><strong>S3</strong></td>
<td></td>
<td>Retro</td>
<td>1 Meeting</td>
<td>PO</td>
<td>Sprint Demo</td>
<td>1-1.5 hrs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Planning</td>
<td>1-1.5 hrs</td>
<td>PO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
<td>1 Meeting with Customer Team</td>
<td>PO</td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
</tr>
<tr>
<td><strong>S4</strong></td>
<td></td>
<td>Retro</td>
<td>1 Meeting</td>
<td>PO</td>
<td>PSI Demo Prep</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Planning</td>
<td>1-1.5 hrs</td>
<td>PO</td>
<td>PSI Demo Prep</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whole Product Team</td>
<td>1 Meeting with Customer Team</td>
<td>1 Meeting with Customer Team</td>
<td>PO</td>
<td>PSI Demo Prep</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** PO stands for Product Owner.
# Sprint Schedule

<table>
<thead>
<tr>
<th>Whole Team</th>
<th>W</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum Master</td>
<td>Retro 1 hr</td>
<td>Standup -15 mins</td>
<td>Standup -15 mins</td>
</tr>
<tr>
<td>Tech Lead</td>
<td>Sprint Planning 1-1.5 hrs</td>
<td>Scrum of Scrum 15 mins</td>
<td>Scrum of Scrum 15 mins</td>
</tr>
<tr>
<td>PO</td>
<td></td>
<td>PO Meeting 2 hrs</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M (Monday)</th>
<th>T (Tuesday)</th>
<th>W (Wednesday)</th>
<th>T (Thursday)</th>
<th>F (Friday)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum of Scrum 15 mins</td>
<td>Scrum of Scrum 15 mins</td>
<td>Scrum of Scrum 15 mins</td>
<td>Scrum of Scrum 15 mins</td>
<td>Scrum of Scrum 15 mins</td>
</tr>
<tr>
<td>Customer Team Meeting 2 hrs *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Customer Team Meeting - Once every other week, on any suitable day

- End of each PSI, whole day for Demo
- Beginning of each PSI, two whole days for Planning

---

Copyright © 2005-14, AgileFAQs. All Rights Reserved.
Retrospective
Observations

What we found.
Our Observations

- Teams are **excited and willing** to try and learn from new ideas.

- Throughout the organization, there is a very early, but growing **culture of openness, communication and transparency**.

- The sense of **team commitment** is improving and team members are willing to jump in and help each other.

- The teams are willing to **stretch** as there is a **sense of commitment** to their Products’ success.

- There is **improved collaboration** within the teams. Also, the POs are interacting with Uber POs and Marketing Reps more.

- There are **varying levels of maturity** along the agile journey amongst the teams. E.g. Apex has a better grasp on agile practices. However, all observations noted apply (to some degree) to all teams that we worked with.

- Teams are picking up the **practices**, but I would like to see them critically think about the **values** behind those practices and figure out variations or adaptations to those practices to better suit their environment without sacrificing the value.
Our Observations...

- **Misconceptions about Agile**: Overall the teams are excited about Agile, however they need a lot of help to make the transition. There are quite a few misconceptions about Agile that need to be cleared.

- **Management Commitment**: There is a strong sense that the management is committed and supportive of Agile.

- **Multiple roles**: Some of the key members on the teams like PO, SM are playing multiple roles, quite thinly spread.

- **Domain Knowledge**: System domain is fairly well understood by the team, however business domains and value propositions need more understanding.

- **Churn and volatility**: Teams agree on PSI Goals, but during the PSI there is a good 40-50% churn.

- **Mini-Waterfall**: At a very high-level, the overall process followed seems like mini-waterfall process.
Our Observations...

- **Team lacks skills:** Most team members lack necessary skills to be successful in an agile environment. They will certainly need a local coach/mentor to help them.

- **Need help with Scrum:** Based on our interactions with the team, help on Scrum framework, relative complexity estimates, planning, stories, collaboration, etc. is needed.

- **Lack XP Skills:** Developers and Testers need specific help with eXtreme Programming skills and practices.

- **Old-school Testing Practices:** Overall testing practice still seems to be in the “inspection” model, rather than building “quality-in” model

- **Distributed Development Challenges:** Distributed nature of depended teams is causing quite a few challenges. Strong coaching is required to easy some of the problems.
Agile Skills Lacking in the teams

- Embracing uncertainty/change & finding effective ways to deal with it
- Tight collaboration and communication with everyone involved
- Eliminating Waste
- Collective Ownership, Drive and Discipline
- Fail-fast: Breaking a large problem down into small safe-fail experiments and then willing to try and learn quickly.
- Systems thinking
- Critical thinking
- Open to experimenting with radical ideas

Copyright © 2005-14, AgileFAQs. All Rights Reserved.
Misconceptions

- Roles of PO, SM, TechLead, Tester
- Estimation, Velocity and Capacity
- Why Agile?
- Self-Organizing Teams
- Clean-code
- User Stories and Acceptance Criteria
Reasons for Churn

- PLMT changes high-level priority
- Lack of collaboration between the teams, the Architects and others stakeholders
- Testing & integration done late in the cycle causing sprints/PSI to overflow
- In some teams the experienced team members are moved out of the team to play other roles leaving a void behind
- In some critical areas, poor implementation is leading to more churn
- Teams are structured architecturally which creates heavy inter-dependencies between them and multiples the churn introduced by the above points
Sample Stories

As a user, our database should support storing of Variety color information against each variety
2 Points - 18 hrs

Tasks:
1. Adding column to variety table to store - 2 hrs
2. Modifying the Business object of variety - 8 hrs
3. Testing - 8 hrs

As a user, I want our setup module "variety" screen should have facility to assign color to individual variety
3 Points - 17 hrs

As a user, I want our map variety layer legend should support variety color information
2 Points - 15 hrs

Acceptance Criteria
* None listed

Acceptance Tests
* None listed

As a User I want to fix defect related with Measurement in our database - I

As a User I want to fix defect related with Measurement in our database - II

As a User I want to fix defect related with Measurement in our database - III

Acceptance Criteria
* None listed

Copyright © 2005-14, AgileFAQs. All Rights Reserved.
Potential Internal Coaches

- PersonX (PO) can be mentored to be good Agile Coaches.
- Due to the short time, we were unable to spend more time with people to identify other potential coaches.
Conclusion and Recommendations
Potential Risk

- **Dependencies**: The way teams are structured and the amount of inter-dependencies between them.

- **User Stories** are very technical in nature, missing the “why” and focusing too much on the implementation.

- Teams are unable or unwilling to retrospect and make intelligent adaptations based on their context.

- **Code Hygiene & fundamental development practices** aren’t in place.

- There is a gap in what the teams think of **ownership** and what is required for a healthy agile team.

- Teams are missing a true **Scrum Master** (process facilitator role)

- Some teams are not pushing back on extra-work load, as a result they might be at the verge of burning out and they are compromising quality.

- There is a lack of **overall product level testing** focus.

- **Risk identification process** for each Epic/Story needs more rigor
Recommendations

- Move to Features Teams.
  - However the teams are not yet ready for feature teams. They still need to build some deep expertise in their modules.
  - Just-in-time teams with a short focus may be a means of bridging the gap.

- Start practicing Acceptance Test Driven Development.

- Find experienced Scrum Masters for the team.

- For coaching: Focus on one product at a time & create real success story from whom other teams can learn.

- Once or twice a year, facilitate a Product Discovery Workshop with the key stakeholders.

- Seriously reconsider how team members are measured.
Developers need help with the following topics

- Good Design Practices
- Understanding and improving code quality
- Understanding the bigger picture and asking for it
- Impact Analysis
- Working effectively with Legacy Code
- Writing effective unit tests
- Test Driven Development
- Working effectively in a distributed environment
- Evolutionary Database Design and Refactoring Databases
- Thin Slicing and Evolutionary + Incremental Design/Implementation
- Fail Fast approach
- Knowledge Sharing
- XP Practices like Continuous Integration, Pair Programming, etc.
Tester need help with the following topics

- Work more closely with the Development Team daily to build quality in and not just test at the end.
- Collaborating with PO and Devs to build domain & tech knowledge
- Collaborating with the PO to better define User Stories and Acceptance Criteria
- Working with Devs to practice Acceptance Test Driven Development
- Exploratory testing and mistake proofing.
- Create and maintain an effective automated test suite
- Enhancing product centricity over just being their scrum team focused
PO need help with the following topics

- Owning the Product Vision and Roadmap (at least feature level)
- Understanding the business drivers, priorities and value chain
- Writing effective user stories
  - Slicing stories functionally rather than technically
  - Defining better Acceptance Criteria for the stories
- Helping the teams independently make certain decisions
- Risk identification and mitigation for each story
- Collaboration with other team members to work-out dependencies
- Facilitating better knowledge sharing with the team
SM need help with the following topics

- Using techniques like Value Stream Maps to identify bottlenecks
- Create Self-Organized Empowered Team
- Emerge as leaders and be the voice the team, shielding the team from external interferences
- Facilitating more effective meeting (Planning Meeting, Demos)
- Capacity Planning
- Story Point (Relative Complexity) Estimation
- Intent behind standup meeting
- Process to following up on Retrospective action items
High Level Plan

1. Understanding Roles and Responsibilities on a Team
   - Getting a simple process structure in place
   - Improving code hygiene & automated testing

2. Establishing a foundational development (XP) practices
   - Focus on one-team to create a success story

3. Spreading the success story to other teams
   - Establish practices across the products (feature teams)

4. Becoming proficient with advanced development (XP) practices
   - Create a product-level success story
   - Get to continuous delivery

Starting Nov
<table>
<thead>
<tr>
<th>1st Quarter</th>
<th>2nd Quarter</th>
<th>3rd Quarter</th>
<th>4th Quarter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting Nov</td>
<td>Initiatives started in 1st Quarter will continue on.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Detailed Plan**

- Feature Teams
- Evolutionary Design

Initiatives started in 1st Quarter will continue on.

- Hire or Coach people to be real Scrum Master.
- Identify & Coach a Pilot Team to create Success story. Teams don’t know what real agility means.

### 1st Quarter

- XP Practices Overview
- Retrospectives
- Acceptance Criteria
- Writing Effective Stories
- Automated Builds
- Standup Meetings

#### Testers desperately need building-quality-in-the-process

- Devs desperately need basic OO Skills training and mentoring
- Capacity Planning
- Story Point Estimation
- Focus on Code Hygiene
- Effective Planning

#### Testers desperately need building-quality-in-the-process

- POs should stop acting like BAs. They should start functioning as real POs.

### 2nd Quarter

- Self-Organized Teams
- Mistake proofing skills for Testers
- Design Bootcamp Training
- Managing Legacy Code
- Slicing Stories
- Value Stream Mapping
- Pair Programming
- Team level Continuous Integration with Automated Tests & Code Quality Metrics

#### Test-first Development

- Test-First Development
- Test Driven Development

#### Continuous Integration with Automated Tests & Code Quality Metrics

- PIs should stop acting like BAs. They should start functioning as real POs.

### 3rd Quarter

- Overall Product level Test
- CI across Product Teams
- Test-First Development
- Feature Teams
- Zero Downtime Deployment
- Design Patterns Training
- Refactoring Legacy Code
- Scrum of Scrums
- Product Discovery Workshops

#### Refactoring Legacy Code

- Managing Legacy Code
- Slicing Stories
- Value Stream Mapping
- Pair Programming

### 4th Quarter

- Self-Organized Teams
- Mistake proofing skills for Testers
- Design Bootcamp Training
- Managing Legacy Code
- Slicing Stories
- Value Stream Mapping
- Pair Programming
- Team level Continuous Integration with Automated Tests & Code Quality Metrics

- Test-first Development
- Test-First Development
- Test-First Development

#### XP Practices Overview

- XP Practices Overview
- Retrospectives
- Acceptance Criteria
- Writing Effective Stories
- Automated Builds
- Standup Meetings

#### Testers desperately need building-quality-in-the-process

- Devs desperately need basic OO Skills training and mentoring
- Capacity Planning
- Story Point Estimation
- Focus on Code Hygiene
- Effective Planning

#### Testers desperately need building-quality-in-the-process

- POs should stop acting like BAs. They should start functioning as real POs.

### Focus on Flow

- XP Practices Overview
- Retrospectives
- Acceptance Criteria
- Writing Effective Stories
- Automated Builds
- Standup Meetings

#### Testers desperately need building-quality-in-the-process

- Devs desperately need basic OO Skills training and mentoring
- Capacity Planning
- Story Point Estimation
- Focus on Code Hygiene
- Effective Planning

#### Testers desperately need building-quality-in-the-process

- POs should stop acting like BAs. They should start functioning as real POs.

### Continuous Deployment

- XP Practices Overview
- Retrospectives
- Acceptance Criteria
- Writing Effective Stories
- Automated Builds
- Standup Meetings

#### Testers desperately need building-quality-in-the-process

- Devs desperately need basic OO Skills training and mentoring
- Capacity Planning
- Story Point Estimation
- Focus on Code Hygiene
- Effective Planning

#### Testers desperately need building-quality-in-the-process

- POs should stop acting like BAs. They should start functioning as real POs.

### XP Practices Overview

- XP Practices Overview
- Retrospectives
- Acceptance Criteria
- Writing Effective Stories
- Automated Builds
- Standup Meetings

#### Testers desperately need building-quality-in-the-process

- Devs desperately need basic OO Skills training and mentoring
- Capacity Planning
- Story Point Estimation
- Focus on Code Hygiene
- Effective Planning

#### Testers desperately need building-quality-in-the-process

- POs should stop acting like BAs. They should start functioning as real POs.
To Achieve this Plan

- PersonX should play the Strategic Agile Coach role.
- More internal coaches should be identified to help him.

- There are just way too many teams for one person to effective coach all the teams. So Carl alone should coach 1 team at a time. To speed up things and to bring more experience to the table, we have 2 options:
  - Option 1: Hire a part-time Agile/Scrum Coach (available 50% time) + a full-time XP Coach to coach one team at a time. Carl can stay at a strategic-level with the selected team & at the same time guide other teams at a high-level.
  - Option 2: Hire a full-time Agile/Scrum Coach + Two full-time XP Coaches to help Carl coach couple of teams at a time. The coaches could spare a few hours each week to coach other teams as and when they need help.

- eLearning can be used to consistently deliver high quality training on the following topics:
  - Composing User Stories
  - Unit Testing
  - Code Smells and Refactoring
  - Test Driven Development
  - Refactoring Legacy Code
  - Design Patterns
Thank You!

- Naresh Jain – naresh@agilefaqs.com