



# Agile Record

The Magazine for Agile Developers and Agile Testers



Requirements in Agile Projects

[www.agilerecord.com](http://www.agilerecord.com)

free digital version

made in Germany

ISSN 2191-1320

August 2012

**issue 11**



# Agile Requirements: Lessons from Five Unusual Sources

by Raja Bavani

Requirements engineering is composed of four key activities – requirements elicitation, requirements analysis and negotiation, requirements specification or documentation, and requirements validation. Requirements elicitation is performed to discover system requirements through consultation with stakeholders. Some of the sources of this discovery could be system documents, domain knowledge and market studies. Requirements analysis and negotiation is there to analyze requirements in details and negotiate with stakeholders, which of the requirements are to be considered. Requirements specification or documentation is to document agreed requirements in a certain level of detail. Requirements validation is performed to review or validate requirements for clarity, consistency and completeness. These four key activities are critical to the success of all software projects, irrespective of the methodology followed. In projects that follow agile methodologies, these four activities of requirements engineering happen in almost all iterations on an ongoing basis. Hence, all team members have a role to play in refining and validating requirements, without which it is extremely challenging to minimize waste and rework in such projects. Here are the five unusual sources and lessons that can help agile teams improve the way they perform requirements engineering.

## Restaurants & Waiters:

During early 2000, a very simple but profound incident happened when I was in an iteration planning meeting with two geographically distributed teams. We were developing a product for a small product engineering vendor, and the product manager was articulating product requirements for the newly started iteration to all team members. I was with my team in a conference room in India listening to him over the phone. Over the first 30 minutes, I sensed a pattern in his approach. He paused after every requirement or a set of related requirements and asked pointed questions to validate the understanding of team members. It helped him not only validate our understanding, but also provide additional examples to strengthen our understanding. From the next iteration planning meet, our team members started articulating their understanding

without waiting for a question from him. That was a simple but profound incident that triggered thoughts about restaurants and waiters. I started observing how waiters in restaurants understand and manage requirements. In this process of learning, I identified the following takeaways:

1. Welcome customers and stakeholders with a smile whenever you interact – even when you are on the phone
2. Listen well to understand the requirements
3. Rephrase or summarize your understanding to get confirmation before you end a conference call or a meeting
4. Believe in your expertise and stay committed
5. Be flexible enough to reprioritize and accommodate changes as long as it is not too late
6. When it is too late, be polite and communicate the impact
7. Be open and ask for intermediate feedback during your interactions
8. Feel free to talk about ‘what else’
9. Value time and money
10. Apologize when things go wrong

## Airports and Flights:

Airports are among the most dynamic places that we experience. In spite of all dynamism of day-to-day life at an airport, one can see how flights are prepared before take-off. Before every take-off, the crew reports on time, passengers are seated, checked-in baggage is loaded, the quantity of food and beverages including special requests is checked, and several other preparatory activities and verification steps are done in order to ensure a safe and comfortable journey. If for any reason a flight has to make an unplanned landing, e.g. to disembark a traveler or to bring someone on board, all stakeholders understand the consequences and there are role holders who are authorized to approve such needs. Besides, every commercial airport has a control tower which has a team



of experts who are coordinating the landing and take-off operations. This involves a lot of coordination in terms of negotiation and facilitation. From this unusual source, I have learned some of the practical lessons, e.g.:

1. Preparation is critical to success. This is applicable to the entire agile team, including product managers or product owners, agile project managers or scrum masters and agile teams.
2. If you arrive late, you miss the flight. In the same way, you can't accommodate late arrival of requirements during iterations.
3. Any delays or changes can only lead to further delays.
4. Change management can never be effective when there is no support from all stakeholders. You cannot blame an individual or a small group of individuals when things go wrong because of ineffective change management.
5. Dependency management in large projects is necessary to avoid wait time and delays. In case of large projects with many related projects, we need a function or a team that plays a role similar to that of the control tower in airports. In most cases, we call it the governance team.

#### **Families and Children:**

Successful families focus on nurturing their children and acting responsibly. By nature, children are curious, genuine and forgiving. They explore and ask questions with no inhibition. Some of the simple things I learned from this unusual source have helped me in doing requirements engineering effectively in agile projects. For example:

1. Children are curious. They ask questions in different ways to understand the things around them. Likewise, agile team members have to be curious. Collaboration without curiosity can become passive over time.
2. Children explore without any inhibition. Agile teams need to develop the habit of self-led exploration.

Qualities such as action orientation, staying curious, and exploration are essential when agile teams collaborate to understand and refine requirements. Without these qualities they give up by making assumptions or waiting until someone delivers clear requirements. In the real world, clarity of requirements increases through team collaboration, and validation of assumptions is mandatory to improve the clarity of requirements.

## Advertise at



### Schools and Teachers:

I have observed schools and teachers for many years. Schools do make annual plans and daily time tables. However, teachers make minor adjustments as they step forward from week to week. Great teachers are excellent communicators. Also they focus on seeing the big picture and believe in continuous improvement. I learned multiple ways or styles of communication from teachers. Also, I learned following-up and following-through from them. From schools and teachers, here are some key takeaways for agile teams in order to ensure effective requirements engineering.

1. Master your communication skills. Explore different ways and styles of communication to understand requirements.
2. The most important thing for teachers is the success of their students. In order to make this happen, they spend adequate time with students and parents, understand their needs and pay attention to coach their students. Just as great teachers believe in their students and care about their success, agile teams need to focus on the success of the project. With this focus, agile teams will do the right things to ensure that requirements engineering activities are done right on time.

### Ant Colonies:

Ant colonies are home to one of the most amazing creatures on earth. Ants are proactive because they gather food during summer for the winter days. They are focused and efficient. They do this by performing one task at a time and reduce task switching. Each individual ant appears to be a specialist in some task. However, ants are capable of performing every task. Qualities such as these inspired me long ago. What I learned from ants includes:

1. When agile teams move from iteration to iteration, they need to anticipate winter days and prepare themselves in such a way that they can minimize rework or mitigate the impact of conflicts in requirements.
2. In addition to being specialists in one area, team members have to be capable of contributing to one or more additional areas. Also, every team member has to be actively involved in requirements engineering activities by participating in activities such as asking questions, thinking about complex test scenarios, active participation in refining requirements, etc.,

### Conclusion:

In the 8th issue of Agile Record, I wrote an article titled, "Distributed Agile: Steps to Improve Quality before Design" in order to emphasize the fact that quality is a journey that starts from the early stages of projects. When we open our eyes and ears to the world around us and learn from unusual sources, we get an opportunity to apply such lessons and understand how simple things make big differences.

### > About the author



#### Raja Bavani

is Chief Architect of Mind-Tree's Product Engineering Services (PES) and IT Services (ITS) groups and plays the role of Agile Evangelist. He has more than 20 years of experience in the IT industry and has presented at international conferences on topics related to code

quality, distributed Agile, customer value management and software estimation. He is a member of IEEE and IEEE Computer Society. He regularly interfaces with educational institutions to offer guest lectures and writes for technical conferences. He writes for magazines such as Agile Record, Cutter IT Journal and SD Times.

His distributed agile blog posts, articles and white papers are available at

<http://www.mindtree.com/blogs/category/software-product-engineering> and

<http://mindtree.com/category/tags/agile>.

He can be reached at [raja\\_bavani@mindtree.com](mailto:raja_bavani@mindtree.com).