

The Day Dan Realized the Power of Kanban!

by Raja Bavani

Introduction

Adoption of new methods or techniques is a learning experience. Project teams adopt new methods or techniques either deliberately or accidentally. Accidental adoption can happen because of difficult circumstances or unexpected events induced by factors such as an unexpected spurt of changes in requirements, issues related to product quality, and so on. During such circumstances, one may introduce changes to current set of processes or techniques and later realize that the new way or approach resembled a popular process or technique. Once, Dan and his team members went through this experience in their professional life. This article about Dan is based on a real life experience.

Dan's Delight

On Monday after the Labor Day weekend of 2002, Dan was on top of the world because Tom, his boss, assigned him one of the most technically challenging projects in the company. Tom was the IT director of a large manufacturing company that supplied electronic utilities to households through retail outlets. Dan was one of his direct reports who played the role of project manager in IT projects.

This project, named DA (disconnected access), was about enabling their technical support team through a state-of-the-art application to support disconnected access to their central databases. Using this application, their support team could reach end users living in remote areas, provide them with necessary support, collect payment, and print invoices even if there was no connectivity to central servers. The goal was to improve customer satisfaction and this was one of the strategic projects of the year under the CEO's radar.

Customer Confidence

Dan and his team of four engineers studied the high level requirements, analyzed the pros and cons of three different architectures, and selected the most optimal one. Considering the business demand and the estimates, they agreed to deliver the application in three months. Alicia was the point-of-contact in the business and she had worked with Dan on a different project a year ago. She had more than fifteen years of experience in her domain. In project DA, she worked closely with Dan and his team in creating and reviewing the user interface design during the initial weeks. She was happy with the suggestions and ideas from Dan and his team and was confident that the team would deliver results.

Iterative Approach

Dan believed in an iterative approach and decided with his team to execute this project in five iterations of two weeks each. As the application incrementally evolved over these five iterations, and the schedule was aggressive, Dan could not demonstrate the finished product to Alicia before the end of the fourth iteration. Alicia had also invited three customer support managers from different regions to attend the demo.

Delayed Demo

No wonder, in each step of the demo, Alicia and the customer support managers started discussing new options and asking Dan whether the user interaction can be improved further. They suggested new ideas which turned into changes to the user interface or validation aspects. After a two-hour demo, Dan and his team came out with a list of 50 items, some of which were new features, some others were changes, and the rest were cosmetic changes and defects.

The Aftermath

Dan had a suggestion – that Alicia and her three managers plan a one-week acceptance testing after feature completion in order to explore all the features and provide him with their feedback. Alicia agreed. After a week of acceptance testing Dan's list had more than 275 items. Alicia was not happy with the results and Tom, Dan's boss, was shocked. Dan's team was distressed that, in spite of their sincere efforts, they had to manage this escalation.

Joint Response

Tom, Dan, and Alicia had an emergency meeting to discuss the situation. The question in front of them was about prioritizing and closing all items in a timely manner in such a way that they:

- a) Address high priority items first and move on
- b) Optimize by grouping related items
- c) Test adequately to avoid failure
- d) Understand daily progress and make course correction
- e) Maximize the number of items closed per week through weekly analysis



www.agileteamacademy.com

Visualizing the Workflow

In his discussion with Tom, Dan stressed on the need to create a visualized workflow using a white board in a common room or work area – even though Tom was against booking a room exclusively for his team. Tom believed in electronic communication, bug tracking tools, and daily status emails. Dan shared his experience on how emails and outdated tools can diminish team work and cause delays. Finally Tom agreed.

The next day, Dan came in early, took a hard copy of prioritized work items, and created a visual board in the meeting room. His team members helped him with their ideas. They created about 40 rows on the board and divided each row into multiple columns. Each column corresponded to the standard work flow – analyze, code, test, verify, build, user acceptance, done. Each row represented a work item with a work item identification number, short description, and other such attributes.

The Pull Factor

From that day onwards, Dan and his team would begin their work day with a crisp meeting in front of the white board. They also ended their work days by reviewing the items on the white board and discussing ways to improve throughput and quality in future. During the first few days, Dan was orchestrating the team on task

assignment. Mostly he was pushing work items or directing his team members. Gradually his team started not only pulling work items, but also logically grouping work items in order to optimize the time spent in coding and testing. Dan appreciated the level of ownership, commitment, and team work among his team members.

Limiting Work in Progress

Based on his experience, Dan believed that mindless context-switching is a waste of time. This happens as a result of keeping many items open, switching back and forth, and losing focus. He coached his team to complete as many tasks on hand as possible before moving on to the next work item. He also shared with them the ill effects of keeping several work-in-progress items. His team members grasped the truth behind his suggestions.

Collaborative Learning

During these tough weeks, every daily meeting or weekly retrospective was a learning experience. Team members shared debugging techniques, domain-specific issues, etc. with the rest of the team to enrich their knowledge and prevent similar issues. Even during the first week they found ways to minimize the number of defects by discussing and implementing simple techniques to improve quality in each step of the workflow.

The Outcome

Eventually, Dan's team became faster and smarter. Dan could see an increasing trend in the completion of work items. Seeing these positive signs, Alicia regained her trust. Within four weeks, the list of pending items had shrunk to 20.

By the end of the fifth week, they only had ten low priority issues on hand. With a schedule overrun of about 6 weeks, Alicia certified the application for user training and implementation.

Reflection and Recognition

A week after the initiation of user training, Tom and Alicia sponsored a dinner for Dan and his team. On the dinner day, Dan convened a 30-minute retrospective with his team. Alicia, Dan, and his team discussed several aspects of the project including what went well, lessons learned, and areas of improvement. Everyone agreed that as soon as they came across innumerable changes and new work items they worked together to find a new approach and it helped them take charge of the situation. The key factors that enabled them deliver results were:

1. Visualizing Workflow
2. Limiting Work-in-Progress and Minimizing
3. Context-Switching
4. Measuring and Managing Flow
5. Nurturing an Open Environment and Making Processes and Policies Explicit
6. Identifying Improvement Opportunities

Both Alicia and Dan agreed that they needed to consider demos to end users and retrospectives at the end of iterations in their projects in order to minimize the spurt of changes and ideas at project completion.

Kanban Knowhow

Years after this incident Dan reads an article titled 'Introduction to Kanban' and later attends a Kanban workshop. He is able to relate the content of the article and the knowledge acquired from the Kanban workshop to his project experience. There are many similarities and some improvement ideas. He feels proud of himself and his team and shares his thoughts with his team mates. He feels proud because he can connect with the essence of Kanban in his approach. He feels proud because his experience and knowledge make him realize the power of Kanban!

With hindsight, he feels that he could have used yellow stickers or a tool to improve effectiveness and optimized the time spent managing the visual board. He could have hired a Kanban coach to help his team in areas such as value stream mapping. However, he realizes that the popularity of Kanban in the software industry has grown only in recent years.

Six months after these happenings and with a promotion at work, Dan continues to execute projects using Agile, Lean and Kanban.

Takeaways

As I said at the beginning of this article, adopting new methods or techniques is a learning experience. I am sure some of you can connect with Dan's experience. Others must have gone through the journey of planned adoption. For the rest, it is not too late to initiate the practice of Kanban. It works perfectly on maintenance, support, and system administration projects. With some experience, you will be able to apply Kanban concepts to development projects too! ■

> about the author

Raja Bavani



Raja Bavani is Chief Architect of Mindtree'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 published papers 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, IEEE Software and SD Times. His distributed Agile blog posts, articles, and white papers are available at www.blogs.mindtree.com/author/raja-bavani and www.se-thoughtograph.blogspot.in. He can be reached at raja_bavani@mindtree.com.