ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

AGILE PROJECT MANAGEMENT FOR ROOT CAUSE ANALYSIS PROJECTS

Sun K. KIM (1), Javier MONT (2)
1: Bayer Healthcare LLC, United States of America; 2: Abbott Vascular, United States of America

ABSTRACT
Root Cause Analysis (RCA) projects face the issue of frequently changing requirements. When a high degree of uncertainty exists in such a project, traditional project management or the linear DMAIC (Define, Measure, Analyze, Improve, Control) may not provide the agility to respond the constant flux of new information. As a Design for X (DFX) initiative, authors applied Agile Project Management, specifically Scrum Methodology, to two RCA projects. The first RCA project used the formal Scrum methodology and the second project used a hybrid approach of Scrum and Lean Product Development. This paper presents observations and learning from this endeavor. The lesson is that Agile may be a good foundation for RCA projects but project managers need to customize the framework based on the nature of their project and teams.

Keywords: design for X, agile project management, scrum, lan product development, root cause analysis

Contact:
Dr. Sun K. Kim
Bayer Healthcare LLC
Operational Excellence
Berkeley
94710
United States of America
kimsunkist@alumni.stanford.edu

1 INTRODUCTION

When a high degree of uncertainty exists in a project, traditional project management (PMI, 2008) may not provide the agility to respond to frequent changes in requirements and the constant flux of new information. Root Cause Analysis (RCA from hereon) Projects fit this category of projects since they contain many unknown risks and their requirements (business and technical) are subject to such continual revisions. Instead of using traditional project management or the linear DMAIC (Define-Measure-Analyze-Improve-Control) process, the authors applied Agile Project Management, in particular Scrum, to RCA projects. This paper presents observations and learning from this endeavor. One key finding is that using Agile has benefits that outweigh its implementation effort. Another point is, in the initial stage of using Agile, the project manager should customize the Agile elements to the specific needs of the project team in order to increase the chance of project success.

Shortcomings of Traditional Project Management

Despite the wide application of the Project Management Body Of Knowledge, known as PMBOK (PMI, 2008) to most projects, Kapur (2005) lists a few reasons why projects continue to fail. Projects lack clear business objectives; project risks are discovered too late; communication between the customer in the project management is not frequent enough; estimated deadlines are not based in reality; management fails to designate a strong project sponsor; qualified project managers are few; scope creep is common; little or no project reviews exist. The authors have sought to find alternative approaches to address these problems.

In a traditional project management approach, the project manager and stakeholders take much time and great pains to find the "right" customer requirements in the initial planning stage. As such, the incentive for the project manager to adhere to the initial plan is strong. In the case that requirements change during the project timeline, as they usually do, the project plan must change accordingly without overburdening the team with additional planning activities. According to PMBOK, changes go through a formal Change Management system. In reality, many organizations' formal change management system is a burdensome bureaucratic process, and as a result, project managers will avoid making any changes if possible. This certainly should not be the case.

2 BECOMING AGILE WITH SCRUM

Authors applied Agile Project Management to 2 RCA projects under the label of Design for X project. Of the many methodologies in Agile domain, we chose the Scrum framework (Nonaka and Takeuchi, 1995) due to its empirical evidence in its effectiveness. (Paulk, 2011)

Scrum is the representative Agile project management methodology for developing software.

The underlying philosophy behind the framework aligns well with the view of design thinking in that they both emphasize iterative work cycles, prototyping and customer-centered design or development (Brown, 2009).

2.1 What makes Scrum work?

Three key attributes—1) Social Commitment from project visibility, 2) concentrated effort, and 3) flexibility to requirement changes—enable Scrum to be highly agile and productive. It is usual that Traditional project management tried to achieve these attributes but with limited success due to emphasis on process adherence and documentation.

Social Accountability from Visibility

The effects of visibility on behavior bring a social commitment system that is unique to Scrum. Much of social commitment is driven by visibility of the project and the awareness that individual contributions are transparent to the team and stakeholders. This transparency (Schwaber and Beedle 2001) is well documented as the Hawthorne effect (Parsons, 1974) and results in vivid accountability to every team member.

Project prototypes, meeting artifacts and ceremonies allow not only the team members to have real-time project status updates but also observers such as customers, management or stakeholders to clearly be aware of what the team has delivered, how they are performing and where they are headed.

At Sprint Reviews, which occurs from every week to 3 weeks depending on the cycle of the project, a project team presents interim deliverables to the Product Owner for approval of the work done and to

stakeholders for feedback. This frequent cadence of project reviews accomplish multiple objectives such as synchronizing information across the project team and stakeholders and more importantly, providing project team's accountability to customers for completing work packages that provide value to them. As one can imagine, inability to deliver the work committed for the sprint can be socially discomforting and this in turn acts as a motivating factor for the team to get the work done.
Scrum deploys various visual tools. Of many, visual kanban emerges as its fundamental tool. A project team uses visual kanban to coordinate and adapt the work of the team during sprints. Figure 1 shows a Visual kanban where each card represents a work package ("story"). The cards move from the "Backlog" to the "To Do", "Doing" and "Done" columns as the work progresses and is completed. This progression displays project updates for both the team and the stakeholders.
On the behavioral side, progress of the story cards on the visual kanban implies the contribution level and performance of individuals at a day-level resolution. In turn, this transparency motivates each member to perform at a higher level than they would without the awareness that the work progress is visible to others. In addition, another side benefit of visual kanban is preventing unnecessary workflow disruptions. Micromanaging project managers who request daily updates at inconvenient times are able to review the visual kanban without interfering team members.

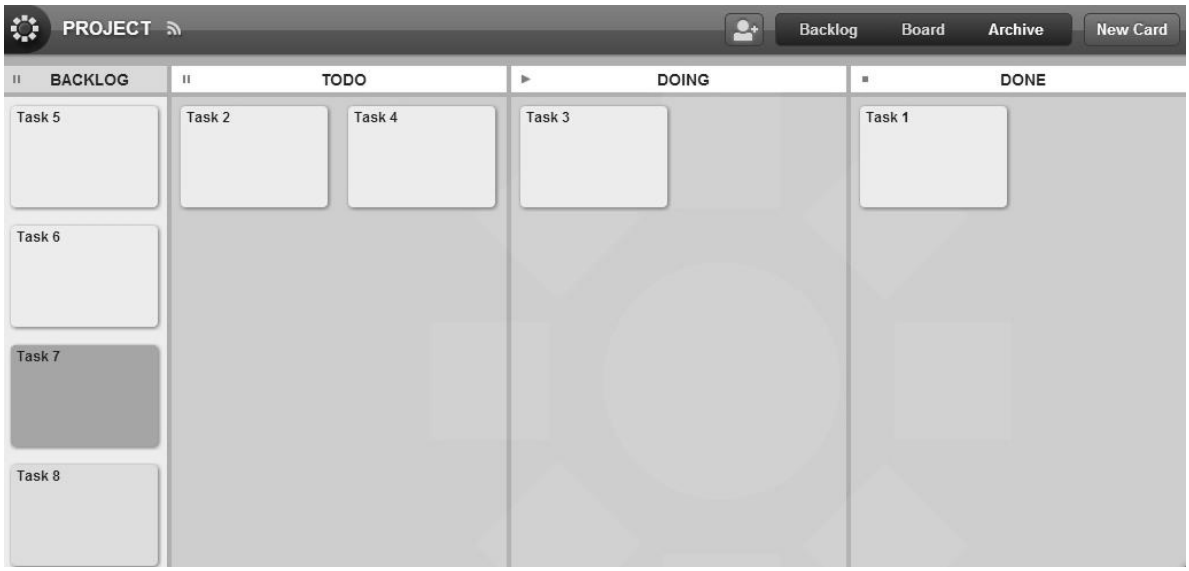

Figure 1. Virtual Visual Kanban (from Leankit Kanban)

The burndown chart (Figure 3) is a visual tool that tracks the velocity of the team (amount of work completed in a sprint) and is used to estimate the completion date for the project.
During Sprint Planning, which occurs before the beginning of each sprint, the team and the Product Owner define and agree on the Conditions of Satisfaction ("Done") for each task or story. The team, based on their availability and willingness, then pulls the stories from the backlog and commits to complete them during the sprint. This is a public announcement of what the team has promised to deliver by the end of the sprint.
Finally, Sprint Retrospective is a ceremony that enhances social commitment to the process and the project. Scheduled at the end of each sprint, it uses the feedback of team members and the Product Owner to improve the process and the project.

Concentrated Effort

The underlying premise of Scrum is that a project team would be fully dedicated to a single project. The reason for this is the inefficiency from working on parallel projects. (Reinertsen, 2009). In reality, employees are tasked with multiple projects at any given time. Scrum reveals the amount of work tasked to the tam. Based on this information, management and the scrum teams can agree on limiting the amount of parallel work. Also not allowing changes during a sprint enhances the focus of the team and minimizes interruptions to its work.
Scrum emphasizes delivering value in time boxes. Authors found that this focus is beneficial in helping teams make progress with a certain cadence because time boxes create artificial deadlines that increase productivity of teams.

Daily standup meetings help the team accomplish their commitments for the sprint. The team decomposes the work required to finish each story during sprint planning. Using the visual kanban board (Figure 2), team members and the Product Owner answer three questions: What did I do yesterday; what will I do today; are there any impediments in my way. This daily social ceremony at the beginning of each business day resets the team members' attention to the project managed by Scrum methodology compared to other projects.

Figure 2. Daily Standup meeting at a physical Visual Kanban board

Flexibility to Requirement Changes

At a macro-level, Scrum allows flexibility because changes of priority (or requirements) are allowed between sprints during Backlog grooming. Because sprint deliveries occur every 2-4 weeks and risk management is reviewed during this time frame, this allows for a great flexibility to respond to any changes in project requirements.

The ability to be flexible also shapes project planning. The team creates rough estimates of the effort to complete all the stories in the backlog. During Backlog grooming, the Product Owner prioritizes stories and the team creates more detailed estimates of these stories. This helps the team focus its planning efforts on stories that have more certainty to be implemented.

Backlog grooming for each sprint assures that the team focuses on completing the highest priority stories. The Product Owner with the input of the team prioritizes the backlog to deliver the highest value to the customer. This is done before each sprint, since no changes in priority are allowed within the sprint. This focus on completing work packages that provide certain functionality or value to the stakeholders at a short frequency reduces the risk of the project. Figure 1 shows a backlog during a Sprint: the highest priority tasks have moved to the "TODO" column and are in various stages of completion. The remaining cards in the "Backlog" column are prioritized with the highest priority at the top.

Daily standup meetings also provide flexibility to the team. The meeting usually uses a board with a list of deliverables to be completed during the sprint. The daily meetings, with its questions, allow the team to adjust to changes and self-organize to achieve them.

In its native form, Scrum was optimized for software development and may not fit the nature of the product development cycle. With minor adjustments however, it provides an efficient project management platform for Root Cause Analysis projects which are subject to frequently changing requirements.

3 APPLYING AGILE APPROACH TO ROOT CAUSE ANALYSIS PROJECTS

The authors formed teams of engineers to apply Agile Project Management for two Root Cause Analysis (RCA) projects. Since no team member was trained to Agile Project Management or Scrum methodology, authors gave a brief overview at the project kickoff meeting and teams applied Scrum soon afterwards.

The first project adopted the formal Scrum methodology to improve the yield of a production line. Based on lessons learned from the first RCA project, second project used a combination of Lean Product Development and selected elements of Scrum methodology of which the goal was to investigate the root causes of a product defect that led to a major production halt. Table 1 summarizes the main differences in the Agile framework between these two projects.

Table 1. Differences in Agile Framework

	First RCA project	Second RCA project
SCRUM CEREMONIES		
Sprint length	Fixed: 2-3 weeks	Variable: 1-2 days
Sprint Planning	Yes	Combined with Sprint review
Daily Standup	Yes	every 2 to 3 days
Sprint Review	Yes	Combined with Sprint review
Backlog Grooming	Yes	Combined with Sprint review
Sprint Retrospective	Yes	One every 2 weeks
SCRUM ARTIFACTS		
Physical Visual Kanban	Yes	Yes
Virtual Visual Kanban, Burndown Chart, User Story points	Yes	No

3.1 Testing the Scrum framework in the 1st RCA Project

The company faced a demand increase for a recently launched product but its manufacturing processes suffered of low production yields. Management set a goal to increase production yields and reduce scrap cost. A "core team" of manufacturing engineers was responsible to achieve these goals and also to address issues that affected the line. This team investigated several failure modes and tracked the status of implementation tasks on a weekly basis. The team reported its progress and tracked production line metrics.

Application of Formal Scrum framework

The core team prioritized the defect types based on their yield and financial impact to create a Release Backlog. The Scrum team was composed of selected members of the core team; they were able to spend an average of 10-25% of their time in the RCA project since their first priority was to support the core team. The Scrum team completed 8 sprints (four 2-week sprints followed by two 3-week sprints) and used the formal Scrum framework and artifacts.

The team applied the A3 problem solving approach (Sobek and Smalley, 2008) within the Scrum framework and used Leankit kanban, a web-based kanban software, to keep the User Story information accessible in an electronic format.

What was Beneficial (and its tradeoffs)

The new ceremonies of the Scrum framework helped the team define goals and focus its efforts to achieve them. Backlog grooming and Sprint planning helped prioritize and define the work to be completed during the sprint. Finishing the deliverables for the Sprint review meeting helped the team focus on the critical activities to achieve them. Although the team members did not like to spend time on these meetings, they became more comfortable with them during the project.

The Daily Standup meetings in front of an Accountability Board helped the team adapt to changes within the sprint. Before using Scrum, team members met informally after their meetings, but without specific agenda. The Accountability Board helped the team focus on the actions needed to achieve the sprint goals. Despite its benefits, the board used in this project had several limitations: its information could only be accessed by going to the board, older information could only be stored by taking pictures of it and it only had a time horizon of 2 weeks. Although Leankit kanban could have been used to address this, the team was concerned with losing visibility of the information and the additional effort to learn and maintain the electronic information.

"Sizing" User Stories with points helped define their scope. To size a story, team members independently choose a number representing the size of the effort required to complete the story and discuss differences until they converge on a size. This tool had a short learning curve and facilitated discussion regarding scope between team members. Differences between high and low point votes helped identify tasks that were not clearly defined in the scope of the story. These new tasks elements became new Conditions of Satisfaction for the story. In typical Scrum projects, teams use story points to estimate the completion time. This is done by dividing the total number of points in the release by the average "velocity" (points completed per sprint). Unfortunately, this was not possible to do in this project because we could not estimate the effort or number of sprints needed to determine the root causes of a failure mode. We could neither estimate the effort to implement countermeasures, since it required knowledge of the root causes. Management was uncomfortable without this information and often requested Gantt chart with timelines.

What did not work well

Extending the duration of the sprint resulted in less productivity. After completing four 2-week sprints, the team decided to increase the duration of sprints 5 and 6 to 3-weeks . The hypothesis was that the velocity would increase by more than 50% (3 week/2 week-1 = 50%) since less time would be spent on Scrum meetings. We found that the velocity did not change with the sprint duration, as shown by the slope of the line in Figure 3. We also noted that in 2-week sprints team members were better able to estimate their availability and there was a sense of urgency.

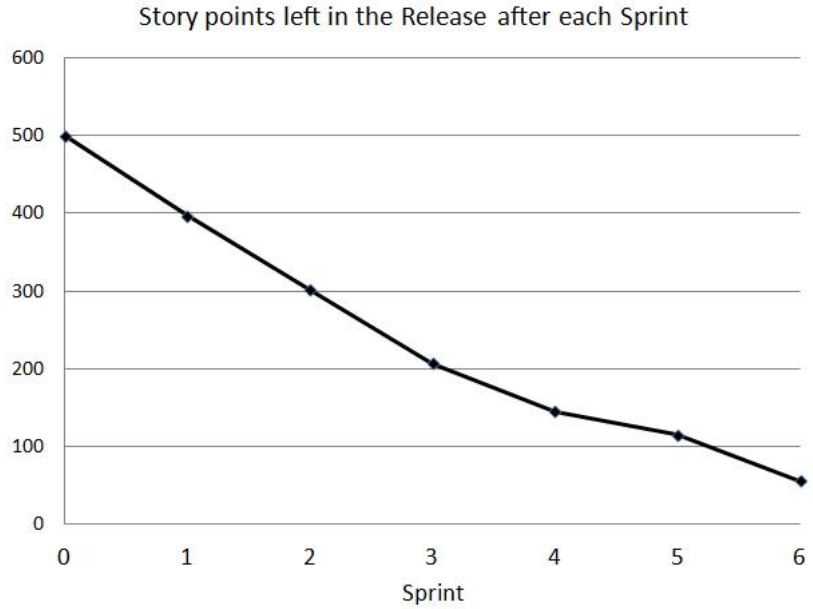

Figure 3. Burndown Chart - The slope of the line in the Burndown chart shows the velocity of the team

Product Owner Approval of "Intermediate" Stories during Sprint Reviews did not add value. In typical Scrum projects, completing stories provides some functionality (or value) to the customer (with the Product Owner as its representative). In this project, stakeholders wanted to know the root causes to implement countermeasures and did not fully understand the value of the "intermediate stories" needed to determine root causes such as development of a continuous measurement system, understanding the inputs to the product and the process with Process Maps, listing of theories with a

Thought map and performing screening experiments. We found that since the Product Owner did not understand the value of these stories, a "Technical Product Owner", familiar with the A3 process, was a good substitute for this process.

What could be improved

After using the Scrum framework for 14 weeks, the team was directed to put on hold the first RCA project and investigate the causes of a new product defect that resulted in a production halt. We used the lessons from the first project to make adjustments to the Agile framework for the 2nd project. Although we had a positive experience with the Scrum framework we wanted to: 1) reduce the sprint duration to less than 2 weeks to maintain the sense of urgency and increase velocity and 2) replace the formal Sprint reviews with stakeholders and Product Owner with reviews when enough progress has been accomplished.

3.2 Adjusting the Agile Framework - The 2nd RCA Project

The investigation lasted for 4 weeks and had a fully-dedicated team of engineers. The investigation was held in a large visual War room (similar to the Obeya tool presented by Morgan, 2006) which improved the communication within the team.

After applying the lessons from the 1st project, the adjusted Agile framework had the following differences with the Scrum methodology used in the first project:

- 1-2 day sprints. Each sprint consisted of an experiment to test different theories and completed an iteration of the PDSA (Plan-Do-Study-Act) cycle (see Figure 4). The team used the results of the current sprint to plan the experiment for the next sprint. The short sprint duration resulted in fast learning cycles, similar to the ones referred by Schipper, et al. (2009).

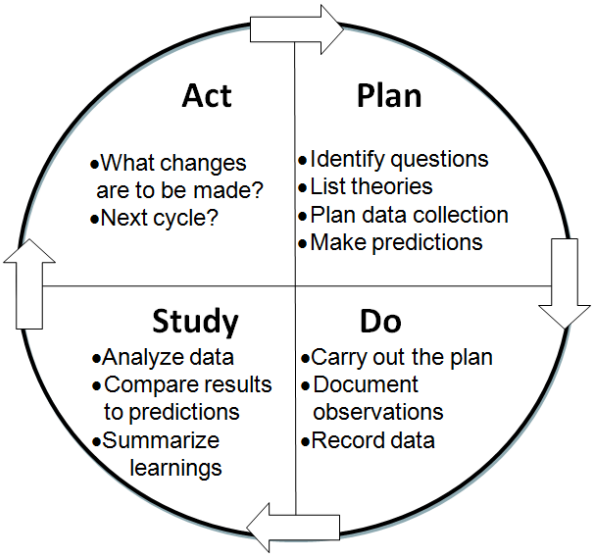

Figure 4. PDSA cycle (adapted from The Scottish Government Health Delivery Directorate, 2008)

- Minimal use of "stories". Since the objective of the sprints was to identify root causes, there was no need to recreate stories. Leankit kanban and story points were not used.
- Consolidation of Scrum meetings. One planning meeting replaced the sprint review, backlog grooming and sprint planning. The review of experimental results was limited to the core team.
- Fewer retrospectives. Retrospectives were held at the middle and end of the investigation.

What was Beneficial

The Visual War room enabled collaboration and quick decisions. Having the information and equipment at a single location (Figure 5 shows a sketch of the room) was a major improvement compared to prior investigations in which there was no dedicated war room where project data resided in electronic presentations and team members worked in their cubicles.

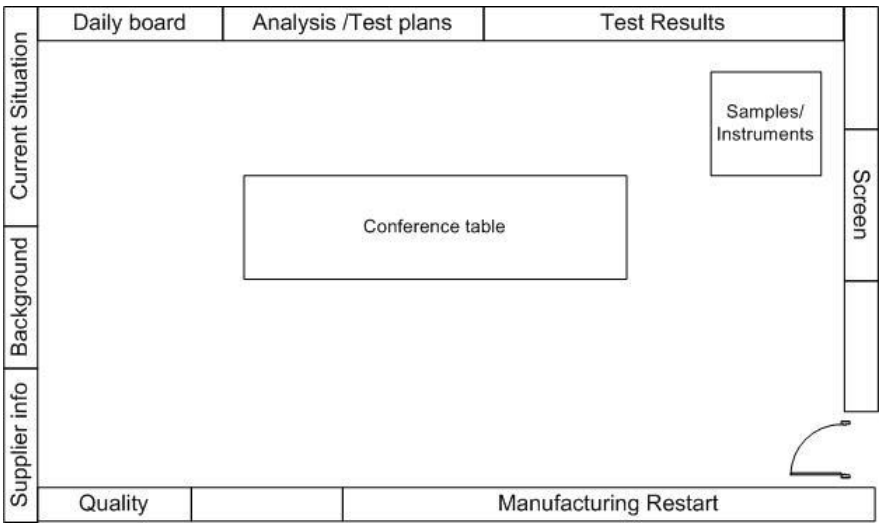

Figure 5. Sketch of room layout

Fast, structured 1-2 day learning cycles. Possible because the team was able to dedicate all their time to the investigation, and had cross-functional support to perform experimental builds in less than a day. The following are other aspects of these learning cycles that worked well:

- Daily morning coordination meetings using the accountability board
- PDSA approach for structured learning.
- Autonomy of the core team. The team decided the theories to be tested, planned the experiment and decided how to carry it with minimal interference outside the team.
- Using the right amount of documentation. The team used flip charts and took pictures to its progress and results (see example in Figure 6).

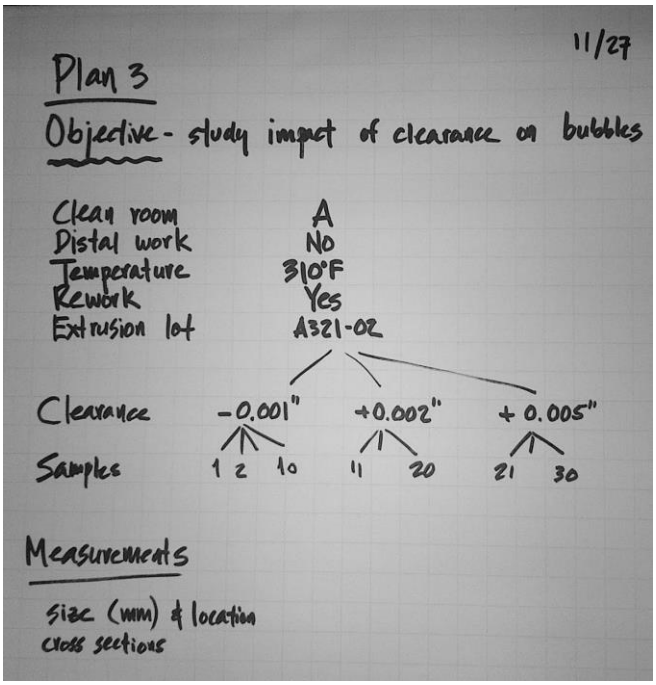

Figure 6. Example of a test plan

What could be improved

The following aspects of the sprints needed improvement:

- Holding "retrospectives" at more frequent intervals. The team held one retrospective during the 4 weeks of the investigation. Making time to perform "retrospectives" on a weekly basis could have resulted in an improved process during the investigation.
- Planning to take advantage of 2 shifts of production. In some early sprints, planning finished at

around 11 am, resulting in 2nd shift building most of the experimental builds. Later sprints started the planning process earlier resulting in earlier builds.

Our experience suggests that Agile or Scrum should be customized according the nature of the project and the team. It is worth noting that all elements of Scrum do complement each other. Removing elements should be done with careful consideration about its impact. For example, in the first RCA project, although the Scrum review was not popular to the team members in the beginning, it became a way to stay connected with upper management and showcase the work of the team later. In another case, one of the authors decided to just list ideas without creating complete full stories (in the format of "As a … I want … so that..." and Conditions of Satisfaction) because creating the stories was perceived as a time consuming process for the length of the sprint (1 week). It became apparent during the sprint that the team did not understand the value of some of the stories selected in the sprint and there was push back from the different stakeholders.

Shortcomings

An aspect of Scrum that does not seem to have an application (at least in investigation or root-cause analysis projects) is the burn-down chart. The total number of points can not be estimated at the beginning, so the chart only shows progress and not estimated completion. Other alternatives (not tested by these authors) are to use a risk burn-down chart or a chart that tracks the difference between predictions and actual results.

4 RESULTS AND CONCLUSIONS

Root Cause Analysis (RCA) projects face the issue of frequently changing project requirements. As a Design for X (DFX) initiative, authors applied Agile Project Management, specifically Scrum Methodology, to two RCA projects to overcome the limitations of traditional project management and the linear DMAIC (Design-Measure-Analyze-Improve-Control) approach. The first RCA project used the formal Scrum methodology and the second project used a hybrid approach of Scrum and Lean Product Development based on the learnings of the first project.

Agile or Scrum is a good start for RCA projects but project managers need to customize the framework based on the nature of their project and teams. Authors observed that elements of Scrum and Lean Product Development complemented each other effectively.

The conclusions of this paper are limited by the small number of projects observed. Availability of resources (part-time vs. fully dedicated) might have had a compounding effect on efficacy of the Agile framework. Future work includes testing this hypothesis and applying Agile Project Management elements to other DFX projects.

ACKNOWLEDGMENTS

The authors would like to thank the RCA Team members for using and providing feedback on the Agile framework; Scott Drumm and Sue Feltovich of Abbott Vascular and Edgar Sur of Bayer Healthcare for sponsoring this project and research.

REFERENCES

Adair, J. (1984) *The Hawthorne effect: A reconsideration of the methodological artifact*, Journal of Applied Psychology, Vol 69(2), May 1984, 334-345

Brown, T. (2009) *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*, Harper Business

Cohn, M. (2009) *Succeeding with Agile: Software development using Scrum*. Addison-Wesley.

Greene, S., and Fry, C. (2008) *Year of living dangerously: How Salesforce.com delivered extraordinary results through a "big bang" enterprise agile revolution.* Session presented at Scrum Gathering, Stockholm.

Mah, M. (2008) *How agile projects measure up, and what this means to you.* Cutter Consortium Agile

Morgan, J. and Liker, J. (2006) *The Toyota Product Development System: Integrating People, Process and Technology*, New York, Productivity Press, p262.

Parsons, H. M., (1974) *What happened at Hawthorne?* Science, Vol 183(4128), Mar 1974, 922-932

Paulk, M. (2011) *On Empirical Research Into Scrum Adoption*,
http://www.cs.cmu.edu/~mcp/agile/oersa.pdf (date accessed 1/3/13).

PMI (2008) *A Guide to the Project Management Body of Knowledge*, Project Management Institute

Reinertsen, D. (2009) *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing
Schipper, T., Swets, M. (2009) *Innovative Lean Development: How to Create, Implement and Maintain a Learning Culture Using Fast Learning Cycles*, New York, Productivity Press.
Schwaber, K. and Beedle, M. (2001) *Agile Software Development with Scrum*, Prentice Hall, 2001. 158 p.
Sobek, D.K. and Smalley, A. (2008), *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*, New York, Productivity Press.
The Scottish Government Health Delivery Directorate: Improvement and Support Team (2008) *The Scottish Primary Care Collaborative - Section 2 - The Model for Improvement*, http://www.scotland.gov.uk/Publications/2008/01/14161901/3 (date accessed 1/3/13).