Managing Intellectual Capital
and Innovation for Sustainable
and Inclusive Society
27–29 May 2015
Bari • Italy

make
learn
TIIM

Management,
Knowledge and Learning
Joint International Conference 2015
Technology, Innovation
and Industrial Management

THE IMPACT OF IMPROMPTU USER STORY MODIFICATIONS ON THE PROJECT, CUSTOMERS AND TEAM MEMBERS DURING A SPRINT

Maureen Tanner
University of Cape Town, South Africa
mc.tanner@uct.ac.za

Martin Lee Pan
University of Cape Town, South Africa
LPNMAR001@myuct.ac.za

Abstract:

Agile methodologies and Scrum specifically, accommodate for changing requirements during the lifespan of a software project. This is typically seen as an advantage but practitioners have reported numerous challenges related to impromptu requirements modifications, mid-Sprint. There is limited academic literature on this issue, which prompts for further empirical investigation. This study therefore investigates how impromptu user story (requirements) modifications impact the project, customers and the team. Qualitative data was captured from a single case study and analysed using the general inductive approach. Some of the findings include: negative impacts on Scrum team members due to conflicts with product owners, and negative perceptions on team members' performance. Product Owners changing and requesting new requirements caused extra pressure on team members to complete work that was not originally agreed upon. Team members did not experience much conflict within the team, but did express discomfort with Product Owners due to the abuse of power.

Keywords: user-story modification, agile, scrum, sprint, teams, information technology, management, innovation

1. INTRODUCTION

As the pace of everyday life increases, computational power grows exponentially, leading to the emergence of many smaller software projects with uncertain requirements. Project managers and teams thus also face more risks and concerns with the delivery of these project. Examples of such challenges include: Not meeting the project expectations due to the difficulty in managing new and changing requirements before project completion; collaboration challenges; development work starting too late leading in poor quality of deliverables (Overhage, Schlauderer, Birkmeier, & Miller, 2011). To keep up with requirements changes throughout the software project, the idea of agile software development was conceptualised about 10 years ago, which further induced a change in software project management and delivery (Cohen, Lindvall, & Costa, 2003).

Agile methods encompasse a process driven, real-time approach to development requirements. Currently, agile methodologies seem to be a trend. According to VersionOne 2013 survey, there was an 11 % increase over the last two years on the number of organisations that agree that agile allows for projects to be completed faster (VersionOne, 2013). In that survey 52% of the respondents reported using agile (VersionOne, 2013).

The most popular form of agile method is Scrum. VersionOne 2013 survey showed that 73% of agile projects followed the Scrum (or Scrum variants) methodology. Scrum breaks the project into small iterations commonly known as Sprints. During these Sprints, some user-stories are scheduled, which are part of the requirements for the end product (Deemer, Benefield, & Larman, 2010; Rising & Janoff, 2000; Sutherland & Schwaber, 2007). These user stories are further broken down in smaller tasks (e.g. designing of the User Interface for that user story) (Schwaber & Sutherland, 2011).

Once scheduled, user stories should not drastically change throughout the sprint to avoid negative impacts on the team's productivity rate and morale (Schwaber & Beedle, 2002). Only minor changes are allowed to the user stories provided that the sprint goal is still achieved. Goldstein (2011) also mentions a few problems that arise with impromptu user-story modification. In particular, he emphasises that it should not happen in the middle of a Sprint as it can cause the Sprint to forcibly be terminated – leading to deflated morale, loss of momentum and administrative time. Agile practitioners have also reported experiencing difficulties in handling impromptu user-story modifications during Sprints. Others believe that such user-story modification during sprints should be avoided altogether as it disrupts the time-line that the project is following (Kochtchi, Witoslaw, & Hager, 2010).

Due to these challenges, several questions now emerge: What happens when changes are not properly managed? How does the team and managers cope with poorly managed requirement changes? This paper attempts to investigate how this phenomenon affects the project, customer and development teams. Some of the reasons leading to these impromptu changes and possible solutions will also be investigated.

Studies have shown that smaller organisations are more likely to adopt agile methodologies and new ideas, because they lack the constraints of organizational history and culture (Vijayasarathy & Turk, 2011). Following this statement, this study has focused on small to medium enterprises (SMEs) and a case study has been conducted in that environment. Data has been collected using semi-structured interviews, observations and field notes and analysed using the general inductive approach (Thomas, 2006).

This study is important as it informs Scrum practitioners of the consequences of such impromptu changes during the sprint and would hopefully encourage them in putting mitigating strategies in place to better shield the teams from such requests. This study also opens up other possible research avenues whereby the suggested solutions to prevent these impromptu changes from emerging could be further investigated.

The paper is organised as follows. The next section discusses the literature review. This is followed by a section describing the methodology employed for the study. The findings and discussion are then presented and the paper is concluded.

2. LITERATURE REVIEW

2.1. Scrum overview

Scrum was first conceptualised by Takeuchi and Nonaka (1986) and originates from the term in rugby where the strategy is to get a ball that is out-of-play, back into play. The name was decided upon because of the similarities that product development had in relation to the actual rugby Scrum – adaptive, quick, self-organising, and have few rests. Elements of Scrum are summarised in Table 1.

Table 1: Elements of scrum

Roles	Product Owner	The Product Owner is responsible for prioritising Return On Investment (ROI) by identifying product features and managing these into a prioritised list (Deemer et al., 2010). The list helps to decide which features the next Sprint will handle. It is possible for the customer to be the Product Owner. Essentially the Product Owner assumes the role of Product Manager, with an added responsibility of team interaction (Deemer et al., 2010; Sutherland & Schwaber, 2007).
	Scrum Master	The Scrum Master leads and manages the Scrum meetings and monitors the progress across the various increments (Rising & Janoff, 2000). The Scrum Master is responsible for the team's efficiency, focus and decision making. The Scrum Master needs to shield the team members from external pressure (Schwaber & Sutherland, 2011).
	Team Member	The team builds the product according to the specifications set out by the Product Owner. A Scrum team is cross-functional – it has all the necessary skills to deliver a working part at the end of each Sprint. The team also manages itself, with a high level of responsibility. The team will decide on what to commit to and how to accomplish that commitment (Deemer et al., 2010, p. 6).
Ceremonies	Sprint Planning	At the beginning of each Sprint a Sprint planning meeting needs to be held. During this meeting, the Product Owner, Scrum Master and team will meet to discuss the requirements of the project. They need to decide on what is important, what needs to be completed and how to go about completing these tasks (Deemer et al., 2010; James, 2010).
	Sprint Review / Retrospective	During the Sprint Review, a demonstration by the team usually occurs so that the customer and stakeholders can see work-in-progress that meet their needs (Deemer et al., 2010). The idea is to get feedback and refine the work in further cycles. The retrospective meeting aims at identifying the problems experienced during the recent sprint, determining whether or not the Scrum team is still on track, and developing ways of improving future sprints (Sutherland & Altman, 2009).
	Daily Scrum Meetings	A Daily Scrum Meeting will happen to ensure the team is making progress. Team members will synchronise their progress and report the details back to the Scrum Master. Team or work related issues will also be raised in these meetings and then reported to the Scrum Master for resolution (James, 2010).
Artefacts	Product Backlog	The Product Backlog is a product road-map and is a single view of everything the team will/could be doing at any point in time, in order of priority (Sutherland, & Schwaber, 2007).
	Sprint Backlog	Once the Product Backlog is ready, the team will break it into sections of work to complete. These sections of work will make up the Sprint Backlog. The team generally orders the Sprint Backlog to match the priority in the Product Backlog (Sutherland, & Schwaber, 2007).
	Burndown Chart	The team needs to constantly monitor their progress and update the amount of time remaining for their respective tasks. Each day they will update their estimate on the time remaining to complete their tasks. These estimates are plotted on a graph known as the Burndown Chart.

2.2. The impacts of story modifications during sprints on projects, customers and teams

There seems to be a lack of academic research into user story modifications, during Sprints. However many practitioner have expressed their concerns and issues related to the phenomenon of user story modifications during Sprints. Some issues that have been identified are: deflated morale, lost momentum and additional administration time - delays in the project (Goldstein, 2011). Table 2 summarises practitioners' experiences with impacts of story modifications, during sprints. Practitioners do not mention much about the impacts on customers, but Bareiss and Katz (2011) indicate customer satisfaction is dependent on the progress of the project. Therefore any impacts on the project will reflect on customer's satisfaction.

Table 2: Practitioners' experiences with impacts of story modifications

Practitioner's Identified Impact	**Affected Party, Person or Object (Project, Customer or Team)**
Deflated morale (Goldstein, 2011)	Team suffers from loss of morale
Lost Momentum (Goldstein, 2011)	Team loses work momentum
Additional administration time (Goldstein, 2011)	Project suffers due to delays
Large modifications incur risk of losing rhythm and consistency (Shimp, 2009)	Project time-lines change, team loses consistency
Team does not complete backlog items on time (Clifford, 2009)	Project suffers due to delays in components
Team spends time on a Sprint and has to re-do the Sprint for the new requirements – wasting time and delaying the project (GoogleGroups, 2010)	Team suffers, because more work is added and time was wasted on the old requirements. The project time-line suffers as the Sprint is re-done

As described above, agile methodologies and Scrum more specifically, accommodates for changing requirements during Sprints. This is usually seen as an advantage but practitioners have stated otherwise (Clifford, 2009; Goldstein, 2011; Maltiriel, 2012). In a search to find academic journals and articles that investigate this phenomenon, one may be unsuccessful as it seems there is limited literature focusing on user-story modifications, let alone modifications during Sprints. The lack of literature with respect to user story modifications, during Sprints, prompts for more investigation into this topic. By identifying the issues, sources and solutions related to story modifications, during Sprints, practitioners can gain awareness of potential and present issues with Scrum, how these issues arise and what can be done to help mitigate the issues – therefore implementing Scrum more effectively.

3. METHODOLOGY

The purpose of this research is to determine how impromptu user story modifications during the Sprint affect the Scrum team members, project and customers. The research therefore used an interpretive and qualitative approach to investigate these. The study focused on a small SME type company, in which a case study was conducted. During the case study, data was collected using multiple semi-structured interviews, observations and field notes. The data analysis process followed the General Inductive Approach in which themes emerged from the raw data provided, in order to provide a succinct and understandable report of the research (Thomas, 2006).

The following research questions were investigated:

- How do user story modifications affect the project, customers and the Scrum team?
- What can be done to mitigate the occurrences of story modifications during Sprints?

A single case study on a SME was deemed sufficient based on Flyvberg (2006) argument that the case study is a very valid tool to collect data and is sufficient enough to generalize about the area.

The respondents interviewed, their position in the team and their employment duration at the time at the study are shown in Table 3. Observations of daily Stand-up meetings with Scrum teams occurred for the duration of the research phase which lasted 2 months. Field notes were taken, while observing areas of concern with relation to projects developments, customer concerns (if any) and team member concerns. The Field notes were used to validate consistency with interview statements and provided insight from a third-person perspective.

Table 3: Respondents' profile

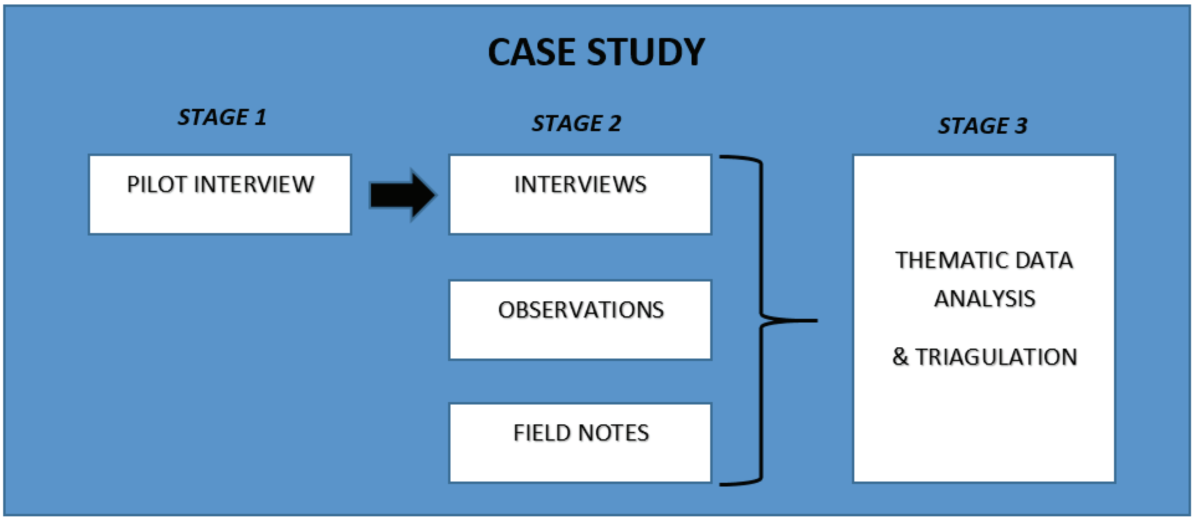
Respondent	Position	Duration of Employment at the given company (roughly)
1	Enterprise Architect	12 months
2	Quality Assurance Engineer	18 months
3	Development Manager	14 months
4	Scrum Master	24 months
5	Senior Java Developer	13 months
6	Product Manager	4 weeks

The following validity processes were adopted, based on recommendations from Creswell (2013):

- The instrument was tested in a pilot interview and initial findings indicated suitability in addressing the research problem and questions.
- Multiple sources of data were used in order to triangulate findings
- The researcher consulted with Scrum experts in the team

A summary of the methodology is provided in Figure 1.

Figure 1: Summary of methodology



4. FINDINGS & DISCUSSION

The main themes that emerged from the findings in relation to the reasons behind impromptu user-story modification and their subsequent impact on the project, the team and the customers are summarised in Table 4. Each of the themes are further described below and compared with literature.

Table 4: Summary of themes

Impact on Project	Impact on Team Members	Impact on Customers
• Delays in delivery and unfinished tasks • Poor Code Quality • Shift in time-lines	• Conflict within the team • Negative perception of team members • Extra Pressure on Team members	• Dissatisfied customers because of project delays • Dissatisfied customers because of poor work standard

4.1. Impact on the Project

The themes pertaining to the impact of impromptu story modification on the project were categorised as: Delays in Delivery and unfinished tasks, Poor Code Quality, and Shift in Timelines. Each of these would result in customers being dissatisfied with the progress of the project as well as team members receiving increased pressure. Each theme are further described below.

Delays in Delivery and Unfinished Tasks

Two of the participants expressed that story modifications during Sprints negatively affected projects as it resulted in project delays. Three participants also mentioned that as a result of these changes, project tasks were sometimes incomplete which further negatively impacted the project progress.

> *"They cause delays in deliverables and result in low quality code" (Respondent 5)*
> *"Sometimes the time-line will shift, mostly tasks are carried over to the next Sprint" (Respondent 2)*

Practitioners' literature also identified that, because of requirements changes mid-sprint, there is a risk of losing rhythm and consistency within projects, of teams not completing backlog items on time and of delays in the project (Clifford, 2009; Goldstein, 2011; GoogleGroups, 2010). Therefore there seems to be consistency in practitioners' literature and this study.

Poor Code Quality

Another consequence of impromptu requirements changes identified by the participants is the poor quality of code being written for the project. The participants mentioned that taking time to finish unplanned or changed requirements prevented developers from completing what was discussed in the sprint planning meetings. Similarly components become delayed due to new requests taking priority over tasks under completion.

> *"They cause delays in deliverables and result in low quality code" (Respondent 5)*

This theme somewhat matches practitioners' literature about not completing items on time, suffering from reduced work capacity or losing consistency (Clifford, 2009; Milunsky, 2009; Shimp, 2009). Incomplete tasks will naturally cause delays in components and lack of good quality code results in tasks being incomplete (according to the requirements), so one can begin to see a cascading effect that will ultimately shift the project time-line.

Shift in timelines

The findings revealed that shifts in timelines occurred due to lack of working components at release dates. The cascading effect was seen throughout the Sprint, starting with poor quality components and ending with a shift in the project time-line. Naturally customers are unhappy with shifts in the project, which results in negative perceptions of team members as it seems they cannot produce quality work on time.

> *"Similarly components become delayed due to new requests taking priority over tasks under completion" (Respondent 2)*

4.2. Impact of impromptu story modification on team members

The impacts on team members were expressed mainly as conflict amongst employees. Team members (developers, quality assurance personal etc.) did not express any dissatisfaction in their teams. However, conflict and tension between Product Owners and team members was apparent. The themes identified pertained to Extra Pressure on Team Members, Negative Perception of Team Members and Conflict within the Team. These are further discussed below.

Extra Pressure on Team Members

Three of the participants mentioned in some form or another, the fact that team members experienced extra pressure from new requirements being added mid-Sprint. Moreover, the Product Owner's abuse of their role created a power gap between team members. Product Owners kept requesting changes in requirements and team members had to abide. Team members feel extra pressure because more work needed to be done while the work that had already been completed seemed wasted.

> *“The effort often outweighs the value so time is wasted on minor issues and the team often feels like they are not doing what was agreed upon initially” (Respondent 2)*

This situation is not ideal as the Product Owners and teams need to work together (Schwaber & Sutherland, 2011). This correlates with Goldstein’s (2011) comment about teams losing momentum.

Negative Perception of Team Members

Four participants mentioned that team members were negatively perceived by customers and management due to the poor workload completion rate induced by impromptu requirements. Product Owners and customers viewed the team members as incompetent due to not completing components or tasks.

> *“The lack of components reflects negatively on the on the team, as it seems they are incompetent” (Respondent 1)*

This theme fits in with practitioner’s literature, recognising deflated morale and losing rhythm and consistency (Goldstein, 2011; Shimp, 2009).

Conflict within the Team

While all of the participants mentioned that team members were satisfied with each other, four of the six participants mentioned that story modifications during Sprints resulted in conflict within the team. Participants also mentioned a high degree of tension between Product Owners and team members.

> *“Team members experience conflict with unknown stories, because they did not agree upon them initially” (Respondent 4)*
> *“Tension is created between PO's and developers and other team members” (Respondent 1)*

Team members also viewed Product Owners as bossy and demanding. Product Owners pushing new requests add extra pressure on team members to meet the changing requirements and finish the original tasks. The negative effects of conflicting teams have been noted by practitioners and can be seen in: deflated team morale, loss of momentum, losing rhythm and consistency and incurring a loss in velocity (Goldstein, 2011; Milunsky, 2009; Shimp. 2009).

4.3. Impact of impromptu story modification on customers

The impacts on the customers were defined as dissatisfaction. Customers wanted to see results, and negative impacts on projects or team members easily cascaded into unhappy customers. Customers also became unhappy if requirements were changed and not well negotiated.

Dissatisfied Customers

Two participants explicitly expressed that customers had been unhappy due to changes in the project components and poor work standard. However, customers were often the source of changing requirements. Product Owners then passed these on to teams. However Product Owners also sometimes misunderstood the requirements, resulting in unsatisfied customers due to incorrect changes being implemented.

> *“Occasionally customers become unhappy, but not often. There was a case where the customer was not happy with the release so the components needed to change” (Respondent 2)*
> *“Low quality code and delays cascade into unhappy customers” (Respondent 5)*

Bareiss and Katz (2011) indicate that customers’ interests are purely in the development of the project. Ensuring that the project meets its deadline is crucial. Therefore customers will be unhappy with incomplete projects or even incomplete progress at each release.

4.4. Possible solutions

The findings revealed that Scrum is perceived as a useful methodology in spite of the negative effects of story modifications which sometimes happen mid-sprint. Participants mentioned that even though Scrum gave rise to issues in projects, it still had more benefits than drawbacks. In this section, some possible solutions to mitigate impromptu story modifications are proposed as follows:

- Ensure the Scrum Master is performing his/her role correctly, the Scrum Master is crucial in facilitating effective Scrum (similar to how a referee must maintain control over players in a game).
- Ensuring that team members and Product Owners work together, not against each other and shielding team members from external forces (customers pushing new requests via the Product Owners)
- There should be no abuse of Scrum roles. Scrum is dependent on roles being played correctly, Product Owners may change or modify requirements, within boundaries.
- The team should effectively make use of Scrum artefacts like Use story points, velocity charts, burndown charts etc. to determine the boundaries with which story modification can occur.
- If a substantial requirements modification is required, it is up to the Product Owner(s) to negotiate with customers on what can be swapped out (something of equal value, in terms of work hours required).
- User story definitions need to be thoroughly defined, validated and managed. Unclear definition and poor validation of user stories can lead to modifications and therefore negatively impact on the project, customers and teams

Throughout the duration of the case study, a lot of changes (as represented by the above recommendations) were being made to accommodate for the weak Scrum process in place at the SME. This indicates an evolving process where an organisation grows and adapts a framework to match its people and processes.

5. CONCLUSION

The purpose and objectives of this research were to investigate the impacts of impromptu story modifications during Sprints, on projects, customers and team members. Possible mitigation solutions were also investigated and proposed. Data was collected through a case study during which semi-structured interviews were conducted.

The primary question was answered in three parts, relative to the project, customers and team members. What was apparent from the findings was the relations each of the three held with each other. Something similar to a domino-effect, by creating a problem in team members it could cascade into the project and the customers soon after.

Impacts on the project were identified as: delays in delivery and unfinished tasks, poor code quality and shift in timelines. Each of these impacts would result in customers being dissatisfied with the progress of the project as well as team members receiving increased pressure.

The impacts on team members were identified as: conflict within the team, negative perceptions of team members and extra pressure on team members. Team members (developers, quality assurance personal etc.) did not express any dissatisfaction in their teams. However, conflict and tension between Product Owners and team members was apparent.

The impacts on the customers were simply defined as dissatisfaction. Customers want to see results, and negative impacts on projects or team members can easily cascade into unhappy customers. Customers may also become unhappy if component requirements are changed and not well negotiated.

Solutions to the impromptu user story modifications mid-Sprint were also proposed. What participants mentioned in their descriptions of how and why story modifications occur, revealed what should be avoided while using Scrum. Hence, the recommendations were based, both on observations and inference from the findings.

This study has provided insight on issues that may accompany user story modifications, in Sprints. This study also opens a door for other researchers to investigate this phenomenon in greater detail. A larger study will add great value to the lack of literature of this topic. Investigation into other SMEs and larger organisations if possible would be useful to determine if the results were isolated or skewed. It would be of great benefit to observe how Scrum differs in a more mature environment and with more Scrum-experienced employees.

REFERENCE LIST

1. Bareiss, R. & Katz, E. (2011). An exploration of knowledge and skills transfer from a formal software engineering curriculum to a capstone practicum project. *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, 71–80.
2. Clifford, J. (2009). Scrum Smells: Going Along To Get Along. *Construx Conversations*. Retrieved 16 July 2012, from http://forums.construx.com/blogs/johnclif/archive/2009/08/05/scrum-smells-going-along-to-get-along.aspx
3. Cohen, D., Lindvall, M. & Costa, P. (2003). Agile software development. A DACS State-of-the-art Report, Retrieved on 21 July, 2014, from http://citeseerx.ist.psu.edu
4. Creswell, J. W. (2002). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research.* Upper Saddle River: NJ: Pearson Education.
5. Deemer, P., Benefield, G., & Larman, C. (2010). The Scrum Primer, Retrieved on 31 July 2014, from http://goodagile.com/scrumprimer/scrumprimer.pdf
6. Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research *Qualitative Inquiry*, (*12:*2), 219–245.
7. Goldstein, I. (2011). Scrum Shortcuts Without Cutting Corners. Retrieved on 31 July 2014, from http://www.scrumshortcuts.com/blog/planning-metrics/sprint-issues-when-sprints-turn-into-crawls/
8. GoogleGroups (2010). PO Allowing Stakeholders to change requirements during Sprint. Retrieved 25 August, 2012, from https://groups.google.com/forum/?fromgroups=#!topic/scrumalliance/7ZK-mJ8SMjg
9. James, M. (2010). Six Pages About Scrum Scrum Meetings. *CollabNet*
10. Kniberg, H. (2007). Scrum and XP from the Trenches. InfoQ Enterprise Software Development Series Retrieved on 31 July 2014, from http://www.metaprog.com/csm/ScrumAndXpFromTheTrenches.pdf
11. Kochtchi, A., Witoslaw, K., & Hager, F. (2010). Changing Story order in Sprint Backlog. Retrieved on 18 July 2012, from http://kunagi.org/iss686.html
12. Maltiriel, K. (2012). Confused about modifying the sprint backlog during a sprint *Programmers Stack Exchange*. Retrieved 15 July 2012, from http://programmers.stackexchange.com/questions/149871/confused-about-modifying-the-sprint-backlog-during-a-sprint
13. Milunsky, J. (2009). TheAgileBuddyBlog. Retrieved 25 August 2012, from http://blog.agilebuddy.com/2009/07/switching-user-stories-mid-sprint.html
14. Overhage, S., Schlauderer, S., Birkmeier, D., & Miller, J. (2011). What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. *2011 44th Hawaii International Conference on Systems Sciences,* 1–10.
15. Rising, L., & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, (July/August).
16. Schwaber, K., & Sutherland, J. (2011). The Scrum Guide, Retrieved on 16 October 2013, from www.*Scrum.org*
17. Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall, NJ, USA.
18. Shimp, D. (2009). Advanced Topics In Scrum. Retrieved on 25 August 2012, from http://advancedtopicsinscrum.com/faq/can-new-work-be-added-during-a-sprint/
19. Sutherland, J., & Schwaber, K. (2007). The scrum papers: Nuts, bolts, and origins of an agile process, Retrieved on 31^{s} July 2014, from https://www.zotero.org/joslash/items/itemKey/NRV6WRHP
20. Sutherland, J & Altman, I. (2009). Take No Prisoners: How a Venture Capital Group Does Scrum, in *Agile 2009*, Chicago.

21. Tanner, M. & Mackinnon, A. (2013). Sources of disturbances experienced during a Scrum sprint, 4th International Conference on Information Management and Evaluation, Ho Chi Minh City, Vietnam, 13-14 May 2013, ISBN: 978-1-909507-20-3
22. Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review*, 137–146.
23. Thomas, D. R. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, *27*(2), 237–246. Retrieved on 31 July 2014 from http://aje.sagepub.com/cgi/doi/10.1177/1098214005283748
24. VersionOne (2013). 8th Annual State of Agile Development Survey. Retrieved on 31 July, 2014, from http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf
25. Vijayasarathy, L., & Turk, D. (2011). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. *Information and Software Technology.*