Practicalities of using Scrum for policy projects
Maryam Mirzaei, Victoria J. Mabin
School of Management
Victoria University of Wellington
Maryam.Mirzaei@vuw.ac.nz, Vicky.Mabin@vuw.ac.nz

Abstract

There is a growing interest in applying agile project management practices such as Scrum, particularly for software development projects. Is Scrum a panacea for all projects? What are the challenges in implementing Scrum outside its dominant application in software development? This paper provides some clues based on the findings of case study research into a project in policy development which has experimented with Scrum. Multiple semi-structured interviews and project documents have been used as sources of data. This data was analysed using the Theory of Constraints Thinking Process (TP) tools to capture the cause and effect relationships that help explain where and why Scrum processes apply. The research identifies two Scrum assumptions that may not hold outside software development namely that the product owner (customer) can gain understanding of the final product only by observing the product itself; and that the project is divisible into small deliverables. In this case study it was found that incremental value is not always feasible, even in projects with higher uncertainty. Nor is incremental value the only way to obtain feedback. The case study also demonstrates how Scrum was subsequently tailored to address specific requirements of this project to address these faulty assumptions. The TP tools were found to be particularly powerful tools for uncovering these assumptions.

Key words: Policy Project, Scrum, Theory of Constraints, Thinking Process Tools.

Introduction:

Project management has been a traditional area of application for operations researchers following the development of CPM and PERT project management methods in the 1940’s. The application of Lean and TOC methods to projects, starting in the 1980’s, have both led to alternative ways of managing projects, namely Agile and Critical Chain project management respectively. Both methods draw on various sporting terms such as Scrum, Sprint and Relay Race to encapsulate two important roles of project management, namely to provide coordination and speed. This paper addresses the use of Agile methods outside their dominant application in software development.

The Agile method such as Scrum differ from CPM and CCPM in two important respects: instead of trying to minimise the duration to complete the entire project as one major ‘deliverable’, Scrum manages the project via a series of Sprints of fixed duration, and each sprint is designed to deliver a number of small deliverables. A daily stand-up
meeting of just a few minutes long, called a Scrum, is used to check each team member’s progress, regroup and adjust as needed.

Since this method has become very popular in software development and IT projects, it is natural that others will try using Agile in non-software contexts. Despite its popularity in Software development Scrum was originally developed by Takeuchi and Nonaka (1986) for new product development. Nevertheless, most materials describing Scrum focus on software developers as their audience. As a result, the Scrum method is left to interpretation in other industries. This paper explores one particular case, in policy development, which has experimented with Agile. The case was one of several explored as part of a larger research project, described elsewhere (Mirzaei, 2015). Data collected from interviews and other sources is analysed using the Theory of Constraint Thinking Process tools to capture the cause and effect relationships that help explain where and why Scrum processes apply.

**Methodology**

This case study involved multiple semi-structured interviews with the project manager. For the purpose of triangulation of data, other sources of information were obtained via 2 meetings with a project consultant and the project manager, participation in 2 workshops conducted as part of the project, the firm’s website, project manager’s website, and online material suggested by the interviewees.

Yin (2003) suggests using cause-and-effect models in case study research. This paper uses three cause and effect modelling tools from TOC thinking processes for analysing data: Goal Tree, Current Reality Branches and Evaporating Cloud. The case analysis began by constructing a Goal Tree. Goal Tree as explained by Dettmer (2011) is a graphical, hierarchical representation of the system giving its goal, critical success criteria and necessary conditions. Information from interview and project documents was used to identify the goal, success criteria, and necessary conditions in this project. The Goal Tree was then presented to the project manager for further clarification and modification. This process confirmed that the Goal Tree represented the intention of the project as it was perceived by the project manager. We then developed Current Reality Branches (CRB) to explorer causes of issues identified that were hindering the goal from being reached. CRB is a sufficiency-based logic tool that provides a snapshot of reality and is used to illustrate causal linkages between actions or polices and their undesirable effects (UDE) in the current situation (Cox, Blackstone & Schleier, 2003). Finally the Evaporating Cloud (EC) was used to resolve a conflict or dilemma between two opposing actions identified in the proceeding analysis. EC is a necessity-based logic tool (Scheinkopf, 1999). EC is formulated in such a way that demonstrates the connection between opposing actions and their common goal. Once the EC is constructed and assumptions underpinning each logical connection are explored, false assumptions are identified, and this process would resolve the conflict. If all assumptions are found to be valid, then ‘injections’ are proposed to invalidate them. Injections are in fact, actions, or conditions that are proposed to resolve a conflict. The TOC thinking processes also have a mechanism for verifying the validity of cause and effect logic trees through eight specific
tests called the Categories of Legitimate Reservation (Dettmer, 2007). These tests guide us to check that, for example, the cause and effect entities and relationships exist, and that they clearly and adequately describe the situation.

**Introduction to the case**

The case project was a policy development project within a major New Zealand’s Government Ministry. The project was initiated as part of the Government agenda on raising the value of exports. The policies were expected to increase not only the volume of products, but also its net value. The latter was expected to be achieved by moving local businesses further along the value stream. The purpose of this project was to investigate how the Government can help businesses to make such a move. It was also expected to identify existing barriers that are stopping companies from taking those opportunities. The result was used to recommend interventions that remove those barriers. The above information was produced in the form of reports. The project produced six sub-reports. The reports contained policy-related recommendations to the Government. The process of developing such recommendations involved gathering and organising information, research, negotiations and consultations. Information was gathered from academic literature from countries similar to New Zealand as well as the broader literature. The information would address all different parts of the value chain. The project also involved negotiations with industry representative groups and consultation with subject matter experts. While all the above processes were similar to past undertakings in the organisation, such undertaking had never been structured as a project. The project manager argued that policy projects are costly, time-consuming and involve great effort, and are amongst the least mature management practices. Application of project management techniques was perceived to positively impact the efficiency of the process of this undertaking.

The project team had decided to use Scrum to manage the project. The project manager perceived Agile as a solution to deal with uncertainties. Despite uncertainties with the scope, it was possible to divide the scope into smaller sub-projects. In most instances, these sub-projects were not interdependent. For example, research and recommendations in each industry were clearly independent of other industries. The process also followed a distinct sequence of work. The project manager called such pieces of work milestones with no dates attached to them. Therefore, some level of decomposition was possible. The majority of activities were negotiated within the team on a daily basis. The whole team worked as one unit. The team used daily meetings for negotiating, problem solving and organising the collaboration throughout the day. Though the project team was working part-time on this project, they were in fact dedicated to this project until its completion and were not expected to engage in any other projects during this process. Each member of the project team was specialised in one area and tend to contribute according to their area of expertise. Nevertheless, the tasks were performed according to the sequence of items on the backlog and irrespective of the roles and availability of human resources. Therefore each team member would pick tasks according to what was needed to be done, even if they had to switch roles at times.
Goal Tree

As shown in Figure 1, three success criteria were identified as most important in this project: ‘producing relevant and useful information’, ‘ensuring that information is used by intended users’ and, ‘efficient use of resources’.

Figure 1: Goal Tree

Producing useful and relevant information has three necessary conditions. As information is an important input to this process, accumulating ‘relevant and accurate information’ was the first necessary condition for producing the right output. Next, the process of analysing was important in itself. Such a process turned raw data, or the input information, into appropriate recommendations. Finally, the effectiveness of such recommendations was dependent upon coherence and alignment of those recommendations with each other and also with other works in the Ministry. This in turn prevented any counter-effective action. All these necessary conditions were expected to be possible by engaging experts in the subject matter. In addition, obtaining strategic guidance from senior management was necessary to align the recommendations with other works in the Ministry.

To ensure that the information will be used by the intended users, it was necessary to ensure ‘end users are engaged with the project’ for which strategic guidance from senior management was required. It was also necessary that ‘right communication method is used’ and that ‘information is ready when it is needed’. These were necessary to ensure that information was relevant and was used.

‘Effective use of resources’ was expected to be achieved by three necessary conditions. First, it was necessary that the ‘right decisions are made about allocation of human resources’. For this, clarity about ‘opportunity cost of resource allocation decisions’ was essential. Second, since the majority of work entailed gathering and processing
information, communication was the key. The efficiency of the team was highly dependent on the efficiency of their communication methods. Moreover, to ensure allocation of required resources, it was necessary that ‘requests for resources are signalled early’.

**Challenges and obstacles in achieving the goal**

A major challenge observed in this project was related to interpreting Scrum in the context of a policy-making project. Scrum advocates delivering incremental value at the end of each sprint being of a short, fixed duration. However, in this project the smallest incremental value that could be delivered took much longer than recommended sprint length. We used Current Reality Branches to depict snapshots of undesirable effects observed as a result.

The first Current Reality Branch depicted in Figure 2 demonstrates. ‘Increased effort’ was an undesirable effect that occurred as a result of using the Scrum method. This contradicted the efficiency success criterion. This additional effort was required to maintain the communication tools that were traditionally practised alongside the tools that were prescribed by Scrum. For example, their existing culture imposed a communication method that was not consistent with face-to-face communication advocated by the Scrum method. It was also observed that the project did not have a Scrum Master; instead it had a project manager who planned and managed the project. He was also involved in managing the execution, in other words controlling the project just as a project manager would do. In fact, reflecting on his role, he did acknowledge that his role was more of a project manager than a Scrum Master. Moreover, while Scrum in software development suggests using ‘user stories’, they have to create a concept equivalent to a user story in the policy project. Many types of undertakings in a policy project require deterministic planning, for example: making appointments with particular people, organising events, conducting workshops or conferences. Such activities were important components of the project and could not be translated into a deliverable that is directly contributing to the final product. Moreover, the language of user stories was not familiar to policy analysts.

Something like ‘working software’ in Scrum requires a type of output that is divisible into distinct deliverables. In software development such deliverables are pieces of code delivering a particular functionality. In this project the deliverable was interpreted as a usable report that contained recommendations at the end of every sprint. Although the report contained several recommendations, arriving at each recommendation was not a distinct process. In fact, within each industry, all information, suggestions, and opinions were required to be used collectively to develop an aligned and supportive set of recommendations. As such, a coherent and holistic approach was essential. Such coherence could not have been achieved if the report was produced over time and in pieces. Therefore, because the product was not easily divisible into small deliverables, adopting Scrum’s concept of incremental value was a challenge.
Another set of undesirable effects were related to long-duration iterations and different sizes of iteration as presented in Figure 3. In order to deliver incremental value at the end of each sprint early in the project, they used sprints without a predefined duration. Instead, the first sprint was estimated to take 2.5 months, while the second would take 1.5 months. This way, the use of short sprints with fixed duration was compromised in order to achieve incremental value. This approach contradicted a fundamental principle of Scrum: iterations that are restricted in duration. The importance of this principle was realised early in the project. The project manager suggested that the absence of a restricted duration extended the milestones. This also shows that deadlines in the distant future did not motivate team members to get work done faster. But as the deadline approached they worked harder. Where there are no hard deadline, the finish date was often postponed. This behaviour echoes the ‘student syndrome’ described in TOC literature. Three undesirable effects were observed as a result of this phenomenon: dates and milestones were extended; there was surge of activities prior to milestones; there was downturn of activity after deadline. The project manager also explained the effect of the political context in which the policy projects operate. He explained that due to short political cycles...
in New Zealand, resources can be taken away and priorities can be changed at any given moment. When priorities change and the project has not yet delivered any output, the effort so far becomes wasted. Thus long duration iterations hindered the benefits of Agile approach.

Therefore, the two sets of undesirable effects described above were related to a conflict resulting from the interpretation of Scrum in a non-software development context. This has been presented as an Evaporating Cloud in Figure 4. Table 1 presents injections proposed here (Inj.1, Inj.2, Inj.4) and an injection observed in the case (Inj.3).
Assumption 1: Fixed duration prevents moving milestones, creates cadence which keeps team motivated, reduces planning requirements, and allows more reflection and learning from performance.

Assumption 2: Receiving and applying appropriate feedback is only possible if there is an actual finished report.

Assumption 3: Smallest incremental value takes much longer than recommended sprint (2 weeks).

Table 1 - Injections to Evaporating Cloud

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Possible injections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Inj.1</strong> Use buffer management to increase motivation and prevent moving milestones&lt;br&gt;<strong>Inj.2</strong> Strategically use planning to reduce overall effort and increases productivity</td>
</tr>
<tr>
<td>2</td>
<td><strong>Inj.3</strong> Senior governance group provides feedback before completion</td>
</tr>
<tr>
<td>3</td>
<td><strong>Inj.4</strong> Produce incremental value as work-in-progress reports or other forms of output</td>
</tr>
</tbody>
</table>

Later in the project, the team adopted fixed duration iterations which created a cadence and eliminated these effects. Long iterations also increased the risk of wasted effort. By implementing injection 3 the team were able to break assumption 2. The Evaporating Cloud above also spells out two assumptions in Scrum: feedback is only possible if there is an actual output and deliverable is divisible into small pieces, each of which can be independently produced and presented independently to the end user. These assumptions were not applicable in the context of policy-making projects. Pieces of information, thoughts, and opinions do not mean anything unless they are gathered together and analysed holistically. Assumption 3 is related to the product owner. In Scrum, it is assumed that the product owner cannot provide valuable feedback unless a workable product is presented to them. This may be because software users are less informed about software than developers. Conversely, in this project, product owners were a governance...
group who were well informed about the process and the output. In fact, the project manager later in the project explained that they were able to get feedback regardless of completion. Moreover, unlike software development projects were the process of achieving output is not as important as the output itself, in this project the process was also important. Following an appropriate and rigorous process was essential to ensure the validity of the output.

Therefore, while obtaining feedback was important, having completed a report was not needed to facilitate obtaining feedback. The Scrum assumption that ‘producing incremental value through completed pieces of work is part of the process for obtaining feedback’ was faulty in the context of this project, as feedback on work in progress was also useful, and as an injection a new concept was adopted in this project: epic. An epic would be considered as a sub-project that delivers a report with a set of recommendations, focused on only one industry. All tasks which delivered a report were defined in a Kanban list. Definition of tasks was broad at the beginning. At the start of a sprint, as much work that could be done in a month would be taken into the Kanban list of that sprint. Later, this work was defined in more detail. More items were added to the Kanban list of each sprint, forming a scope buffer. This in turn prevented the team from not having enough work in case of variations. At the end of each sprint the team reflected on its performance and made decisions for the next sprint. Moreover, planning requirements, both at team and individual level, were reduced as a result of adopting fixed duration iterations. This was because all appointments and arrangements could be made well in advance. Such reduced effort for the same output increased efficiency. In addition to the rhythm created by having fixed duration iterations, they also found retrospectives to be a useful process.

A project management approach was developed that was tailored to the requirements of policy-making projects. The method used concepts from Lean, traditional project management and Scrum. This new approach suggests having four major phases for a policy project. They used tasks (for more deterministic activities, both short term and long term), and objectives (for less deterministic activities to be undertaken in the longer term), as units of planning. A major difference between Kanban as was used in this project and the backlog in Scrum is that in this project they had a list of tasks while backlog is ideally a list of deliverables. They also adopt monthly Scrum planning, regular stand-ups along with the Kanban list of tasks.

<table>
<thead>
<tr>
<th>Scrum in software development</th>
<th>Scrum in this project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope is defined in form of a backlog of user stories</td>
<td>Scope was defined in form of a list of objectives and task</td>
</tr>
<tr>
<td>Delivers a pieces of product at the end of each sprint</td>
<td>Completed a list of tasks in each sprint</td>
</tr>
<tr>
<td>Face validity of the output is the key</td>
<td>Process validity was important</td>
</tr>
<tr>
<td>Scrum master acts as a facilitator in a self-organized team</td>
<td>Project manager, planned future events, but also facilitate mutual adjustments of the team</td>
</tr>
<tr>
<td>Events such as sprint planning, and daily stand up meeting used to coordinate the team</td>
<td>Events such as sprint planning, and daily stand up meeting used to coordinate the team</td>
</tr>
<tr>
<td>Face-to-face communication is prime</td>
<td>Formal communication was required alongside face-to-face communication</td>
</tr>
</tbody>
</table>

Table 2 - Snapshot of Scrum process in this case project
Conclusion

Scrum “challenges assumptions of the ‘traditional, sequential approach’ to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines in the project.” (Padda, Arora, Gupta & Sharma, 2014, pp 91). Nevertheless, Scrum too make some other assumptions. A finding of this case study indicates that advocating incremental value is based on an assumption that the product owner knows less than project team about the final output and can gain understanding only through observing the actual product itself. Another assumption of Scrum is that the product is divisible into small pieces. However, this research found that incremental value is not always feasible, even in projects with higher uncertainty. Nor is incremental value the only way to obtain feedback, particularly outside software development.

This case also identified that two major features of Scrum—cadence and delivering incremental value—each served different purposes and were enabled with distinct practices. Cadence improved efficiency and reduced planning complexity, and was achieved by observing events at regular intervals. Incremental value reduced risk and improved communication.

Reference: