The Dilemma of Tool Selection for Agile Project Management

Gayane Azizyan
Ericsson AB
Stockholm, Sweden
gayane.azizyan@ericsson.com

Miganoush Magarian
SAP Innovation Center
Potsdam, Germany
miganoush.magarian@sap.com

Mira Kajko-Mattsson
KTH Royal Institute of Technology
Stockholm, Sweden
mekm2@kth.se

***Abstract*—Even if agile project management tools grow in number and complexity, companies still face difficulties in selecting tools that fit their needs. One such company is Company A, a multinational company that has experienced a great need for a tool supporting their multi-layered management of requirements and projects. For this reason, they commissioned us to perform a detailed analysis of their tool usage needs, placing much stress on adherence to their processes and all the roles involved. In this paper, we present our journey towards selecting the right tool for Company A and the problems encountered when trying to reach our goal. The tool selection process was based on (1) a study of tool features such as usability and extensibility, and (2) observation of the company's process and elicitation of the company's tool usage needs. Our results show that even if there are many tools on the market, it was still difficult to find an appropriate tool. The reasons were the following: (1) a study of general tool features could not provide enough information to account for the company's needs, (2) even people possessing the same role in the company prioritized different features differently; hence, it was difficult to utilize their feedback for selecting a tool, and, finally, (3) even after having observed the company's process, it was still difficult to find an adequate tool. The tools simply could not support the company's multi-level requirements management process. Moreover, they suffered from poor usability as well as imposed their own terminology and process. Overall, the agile tools studied were not a good match for supporting the company's agile project management needs.**

Keywords-components; agile; tool; process; adoption

I. INTRODUCTION

Agile methods sneaked into Company A's overall processes in an unofficial way about four years ago. No official decision for introducing them was made. Neither was there any process facilitating its introduction. Different teams simply started using agile practices on their own, mainly for implementation-level activities, such as tracking the status of tasks. With time, however, even management started using agile practices for managing and tracking requirements. As tool support, they mainly used MS Excel, Word, and PowerPoint for storing and managing requirements and product backlogs and MS PowerPoint for managing projects. The development teams, on the other hand, used simple physical tools such as paper, sticky notes, and whiteboards.

As the use of agile methods grew at Company A, the company experienced that the simple tools were insufficient for and unsupportive in managing large numbers of requirements and projects. Hence, Company A expressed a great need for a better tool. For this reason, they commissioned us to help them with agile tool selection. Our task was to perform a detailed analysis of their tool usage needs where great stress would be placed on adherence to their processes and all the roles involved.

In this paper, we present the results of our attempt to find an appropriate agile project management tool to meet Company A's tool support needs. The company wishes to stay anonymous in this paper, and therefore, we use a fictitious name, Company A, when referring to it. The remainder of this paper is organized as follows. Section 2 presents the research steps conducted during our study. Section 3 gives a brief description of the company. Section 4 lists and describes the criteria used for evaluating currently existing tools. Section 5 presents the tool evaluation results. Sections 6 and 7 describe the company needs based on our observations of its processes and current state of practice. Finally, Section 8 concludes with final remarks.

II. RESEARCH STEPS

Our research was conducted in six main steps: (1) *Literature study*, (2) *Company need identification*, (3) *Tool selection, (4) Tool evaluation,* (5) *Company observation,* and (6) *Analysis of results.*

During the *Literature study* step, we went through the existing literature dealing with tool evaluations and adoptions in agile contexts. Although we looked through many scientific and non-scientific sources, we did not find any objective, detailed evaluations of agile project management tools. The articles we found were limited to high-level discussions of classes of agile tools to be used in different team types [1], tool evaluations aimed at meeting needs of some specific company [2], and tool evaluations solely focused on open-source tools [3]. Other resources included lists of agile tools, as well as some general discussions of their usage [4]-[6]. The *Literature study* step resulted in awareness that our study was unique. We could neither relate it nor base it on somebody else's results and experience. Based on our findings in this step, we understood that companies met their agile tool needs either

 ISBN: 978-1-61208-230-1

by developing a custom tool [7][8], or by selecting an existing tool primarily based on factors such as the tool's popularity [9][10].

In the *Company need identification* step, we studied Company A and its agile environment. Our goal was to get acquainted with the company's process and tool support. To achieve this, we studied the company's processes, projects, and product documentation, as well as had a series of meetings with three company representatives possessing the roles of a Project Manager, a Scrum Master, and a Line Manager. This step resulted in the identification of a preliminary list of high-level tool support requirements and provided a basis for defining tool properties. To ensure that we had an exhaustive list of such properties, we made an additional literature study during which we elicited requirements that were imposed by Scrum practices [2]. Altogether, we collected 21 properties. We presented them to the company representatives and got them accepted as tool evaluation criteria to be used in the *Tool evaluation* step. The criteria are presented in Section 4.

In the third step, *Tool selection*, we looked through agile tools available on the market and selected six tools for a detailed study. The selection was influenced by the following criteria: (1) the company's interest in particular tools, (2) tool popularity [6][11][12], (3) support for the agile methods used at the company, (4) deployment options, (5) availability of integration options with other systems, (6) licensing, and (7) supported platforms. The selected tools were *VersionOne* [13], *Scrumworks* [14], *Rally* [15], *Scrum Desk* [16], *Silver Catalyst* [17], and *Agilo* [18]. Table 1 briefly summarizes their properties.

The *Tool evaluation* step was conducted in two sub-steps. First, the authors of this paper made their own evaluation of the selected tools. The goal was to rate the tools according to the selected criteria and gather an in-depth understanding of how well the chosen tools supported the criteria. The results of this sub-step would then be matched to the company's tool needs to be elicited in the next sub-step, during which the company representatives weighed the criteria by assigning a quantitative measure of how important each criterion was for the company's needs. A range of 0-10 was used for rating both the tools and the criteria. We chose this wide range of values in order to achieve richer granularity in the tool evaluation results, and to have more options for making our choice

In the first sub-step, when doing our own tool evaluation, we used demo versions of the selected tools. Here, we studied their documentation and executed sample test projects. We then individually rated each evaluation criterion for every tool. Finally, we discussed the ratings, removed all types of inconsistencies and ensured that we reached agreement on every rated value.

After reaching consensus on the rated values, we calculated the total and total normalized ratings for every tool. Here, as given by (1), we first summed up the rating u_i^t of each tool *t* over all criteria to yield a total rating U^t for each tool *t*. Further, as given by (2), we divided the total ratings by the number of criteria (*N*), to yield a total normalized rating U_N^t.

$$U^t = \sum u_i^t \quad (1)$$

$$U_N^t = \frac{U^t}{N} \quad (2)$$

The first sub-step resulted in a list of ratings for each criterion, which was used as a base for the evaluation in the second sub-step.

The second sub-step was conducted in form of interviews, with seventeen company representatives possessing the roles of *Developer, Scrum Master, Designer, Agile Project Manager, Program Manager, Product Manager, Development Manager,* and *Line Manager*. During the interviews, we first presented and explained the list of tool evaluation criteria. The representatives then assigned their weights to each criterion.

After having collected all the data from the company representatives, we summed up the assigned weights to

TABLE I. SUMMARY OF THE EVALUATED TOOLS

Tool name	Methods	Deployment	Integration	License	Platforms
VersionOne	Scrum, XP, DSDM	Hosted, Local	Connectors for a wide range of development tools. [19]	Commercial	Windows, Linux, Mac OS X
ScrumWorks Pro	Scrum	Hosted, Local	Bugzilla [20], JIRA [21], Eclipse IDE[22], MS Excel [23]	Commercial	Windows, Unix Mac OS X
Rally	Scrum	Hosted, Local	Connectors for a wide range of development tools. [24]	Commercial, Free	Windows, Linux, Mac OS X
ScrumDesk	Scrum	Hosted, Local	Microsoft Team Fuoundation Server[25], Mantis [26]	Commercial, Free	Windows
SilverCatalyst	Scrum, XP, FDD, Kanban.	Hosted, Local	SVN [27], Trac [28], Wikis	Commercial, Free	Windows, Linux, Mac OS X
Agilo Pro	Scrum	Hosted, Local	Trac, SVN, Eclipse IDE	Free	Windows, Linux, Mac OS X

 ISBN: 978-1-61208-230-1

calculate average weights $U_N^t\ w_{c_i}$. The goal was to calculate the final rating of each criterion c_i of tool t according to (3). Here u_i^t is the rating assigned by us in the first sub-step.

$$r_i^t = \frac{u_i^t \times w_{c_i}}{10} \quad (3)$$

After studying the results, we realized that the second sub-step proved to be inadequate for appropriately identifying the company's needs. First, we observed that different roles and personalities had great impact on the assigned weights. Even interviewees with the same role provided completely different ratings. Second, due to their narrow view of the company's process, and due to their unawareness of the overall tool requirements, the interviewees had trouble in quantifying their needs by means of numbers.

All this made it meaningless to calculate averages. We became aware that this step would not help us in identifying the company's needs to be fulfilled by the selected tool. A deeper analysis was required. This had led us to the creation of the next research step, the *Company observation* step.

During the *Company observation s*tep, we conducted a detailed on-site observation of the company's agile process. The observation lasted for two months, during which we observed the process, conducted interviews, and studied the company's product documentation. All these tasks were performed in parallel. We followed the executed process and attended several meetings, such as Scrum of Scrums meetings, daily meetings, and management meetings. We conducted interviews with the company representatives, the same representatives that were involved in the *Tool evaluation* step.

The interviews took the form of informal discussions that were guided by a semi-formal and semi-structured questionnaire. We chose this over a more formal, structured approach because we felt that most of the interviewees did not have a clear a priori picture of the setbacks and hindrances in their daily work. Hence, informal discussions served better for studying the daily work, identifying pain areas, and eliciting wishes for improvements. The questionnaire used in this step is presented in Table 2. Last but not least, we studied the internal company documents, such as the product backlog, documents describing new requirements, process documentation, and the like. This had helped us to improve the quality and effectiveness of our observations and interviews.

Throughout the *Company observation* step, we continuously analyzed the gathered information and drew out a diagram of the existing process, its inputs, outputs, the roles involved, the tools used, and the difficulties in the tool usage. As a result of this step, we identified six main tool support needs. These six needs served as input to the final step, *Analysis*. Here we tried to match these needs with the features of the evaluated agile tools in an attempt to find adequate tool support for the company. The results of the *Analysis* step are presented in Section 7.

TABLE II. COMPANY INTERVIEW QUESTIONNAIRE

1	How do you work with the existing system?
2	What do you like about the existing system? What works well for you?
3	What pain areas do you see in your daily work? What is inconvenient or annoying for you?
4	Is there anything you would like to have changed or improved? How?
5	Is there anything you would like to have added to the tools you are using? Any specific features or capabilities?

III. COMPANY DESCRIPTION

Our case study was conducted at Company A, which is a large software development company with a complex structure. The department we collaborated with is spread over three locations: Stockholm in Sweden, Shanghai in China, and Rijen in Holland. The different sites collaborate on one large project. The company is hierarchically structured into six different nodes working on different product parts where each node is further divided into different teams spread over several locations. In total, the department has 85 people that are distributed over eight teams, out of which four are situated in Stockholm. It is these four teams that were involved in our case study.

Company A, just as any other company, has many different roles that are involved in management and development. By a "role" we mean a set of responsibilities that are assigned to an individual or a group of individuals. Below, we list and briefly describe the roles that are of interest to this paper. We would like to point out, however, that some of these roles lacked formal definitions at the company. For this reason, we describe them just as we had understood them. In our descriptions, we only list the responsibilities that lie within the scope of this paper.

- *Program Manager* responsible for the operational steering of activities within a program and accountable for deliveries. This role is also responsible for promoting agile ways of working, for release planning, keeping progress visible, as well as planning, assigning and following the program budget.
- *Solution Product Manager (SPM)* responsible for strategic planning, pricing, commercial packaging, marketing, and setting product and professional service requirements. This role acts as a business builder and maintains a competitive position for the product.
- *Solution Architect* responsible for ensuring that the product is scalable and reusable.
- *Agile Project Manager (APM) responsible for prioritizing* and managing the product backlog.

- *Release APM role* corresponding to an APM with the added responsibility of having an overall understanding of the product specifications, as well as presenting the product to the business owners.
- *User eXperience Designer (UXD*) corresponding to any person qualified in User eXperience Design [29].
- *Developer* corresponding to any development team member.
- *Scrum Master* responsible for maintaining and steering the Scrum process.
- *Line Manager* responsible for a particular product line.

IV. EVALUATION CRITERIA

The twenty-one criteria that were used for evaluating the tools are the following:

- ***Extensibility*** referring to whether the tool can be modified or extended. Here, we evaluated whether a tool provided access to the source code, and whether it was offered on a commercial or open source license.
- ***Usability*** concerning the general usability of the tool. Here, we rated the tools solely for their ease of use, also taking into account whether it was necessary to study tool documentation and tutorials.
- ***Connectivity*** describing the connectors, or plug-ins, provided by the tool vendors such as Integrated Development Environments (IDEs), bug-tracking systems or traditional project management tools. Here, we evaluated the availability of such connectors; we also took into account both their number and variety.
- ***Searching*** referring to the searching capabilities of the tool. Here, we evaluated the availability of searching options, taking into account the searching factors.
- ***Grouping*** standing for the capability to group items in a product backlog. We evaluated whether the tool enabled grouping of product backlog items.
- ***Simultaneous editing*** implying whether multiple users could simultaneously edit the same artifact in the tool. While this might seem a basic requirement for tools, we still consider it mainly due to the absence of such options in basic tools such as spreadsheets used for storing backlogs.
- ***Story status tracking*** referring to the opportunity to track the status of a user story. Here, we evaluated whether the tool allowed to record progress of the story. The status could simply be represented as a string,
- ***Group status tracking*** enabling grouping of product backlog items. We evaluated whether it was possible to track the status of the group.
- ***Overall status tracking*** referring to the options of viewing the overall project status. This could imply a high-level summary view of the Sprint backlogs, or a chart showing the number of completed product backlog items over time. For this criterion, we evaluated whether the tool provided feedback on project status and what kind of status reports it generated.
- ***Sorting/Filtering*** standing for the sorting and filtering options. Here, we evaluated whether the tool provided sorting and filtering services and whether it could sort by several criteria and filter by typing in keywords.
- ***Sprint backlog*** dealing with the ability to create and manage a Sprint backlog. Here, we evaluated whether it was possible to prioritize and order Sprint backlog items.
- ***Estimation*** concerning the estimation ability of user stories and tasks. For this criterion, we evaluated whether it was possible to enter estimations of user stories and tasks, and how flexible the estimation measures were.
- ***Stories.*** Here, we evaluated the options offered for creating and describing user stories. Specifically, we evaluated whether it was possible to group stories into epics.
- ***Tasks*** referring to the tasks required for implementing a user story. We evaluated whether it was possible to break down user stories into smaller tasks in a Sprint backlog.
- ***Testing.*** Here, we evaluated the support for testing tasks. This could be realized in form of connectors to testing tools or by enabling the creation of special testing tasks.
- ***Teams*** referring to team management. For this criterion, we evaluated whether it was possible to create teams within the tool, assign team members, and make changes to the team capacity over time.
- ***Planning*** covering the ability to support Sprint planning, that is, selecting items from the product backlog and entering them in a new Sprint. We evaluated whether this was possible, and whether the capacity of the created Sprint was automatically matched to the team assigned to that Sprint.
- ***Progress*** referring to the status of the story or task on a greater level of detail. Here, we evaluated whether the tool enabled entering the amount of work performed on a story or task, and whether it was also possible to see how much work remained.
- ***Board*** covering the provision of a virtual task board for storing user stories and tasks. Here, we evaluated the availability of a task board, and, if so, whether it was interactive and whether it was possible to drag and drop tasks and user stories on the board.
- ***Burndown*** describing whether the tool included Sprint burndown charts, whether they were updatable, and how visually clear they were.
- ***Remote workplace*** relating to the opportunity to access a tool remotely. Most often, this implies that a tool needs to be deployed as a web application so that the user can access the application even outside the office network. Here, we evaluated whether such an opportunity was available.

V. TOOL EVALUATION

In this section, we present the tool evaluation results. We first present the results of our evaluation according to the twenty-one criteria described in the previous section. We then describe the elicitation of the company's tool needs and motivate why it became unsuccessful.

 ISBN: 978-1-61208-230-1

V.1 Our Own Tool Evaluation

As a result of our evaluation, we discovered that all the evaluated tools focused on team-level rather than management-level aspects. The tools included features such as advanced virtual task boards for facilitating development within the teams, but provided only rudimentary requirements and project management support. Moreover, tools covering more features were complicated to use.

The results of our tool evaluation are summarized in Figure 1 and Table 3. The evaluation had led us to a number of conclusions. First, we noticed that the tools were targeted towards Scrum Masters and development teams rather than managers. Five out of the six tools studied provided virtual task boards, as well as possibilities to break down user stories into more detailed tasks and functions for storing and managing the tasks.

The second, more important conclusion concerned the usability of the tools studied. Tools with higher usability offered less features, and tools which provided many features had a notably lower usability. This can be seen in the pivot chart shown in Figure 2 where the ratings of *VersionOne* and *Silver Catalyst* are presented. On average, *VersionOne* provides more features such as Connectivity, Grouping, and Group status tracking, while *Silver Catalyst* has a rating of 0 for several criteria. However, *VersionOne* has a usability rating of 5, which is much lower compared to the rating of *Silver Catalyst,* which is 10. It took us great effort to get accustomed to working with *VersionOne* due to its many features and customization options. *Silver Catalyst*, on the other hand, had a very simple and intuitive interface that was pleasant to use.

V.1 Elicitation of Company's Tool Needs

During the elicitation of the company's tool needs, the company representatives assigned weights to each criterion. After studying their feedback, however, we had to reject all the results achieved in this step, for the following reasons. First, it became clear that even interviewees with the same role provided sometimes contradictory ratings. Second, due to their limited view of the company's process and unawareness of the overall tool requirements, the interviewees had trouble in quantifying their needs by means of numbers.

Since the ratings provided by the company representatives strongly varied, it is meaningless to show them all in this paper. For illustrative purposes, however, we show three sample responses in Table 4 and Figure 2. Table 4 shows the weights and average weights assigned to all the criteria by three different Scrum Masters whereas Figure 2 shows a pivot chart comparing the two contradicting weights assigned by two different Scrum Masters.

Looking at the presented values makes it obvious that different people possessing the same role assigned weights in radically different ways. For example, for the *Progress*

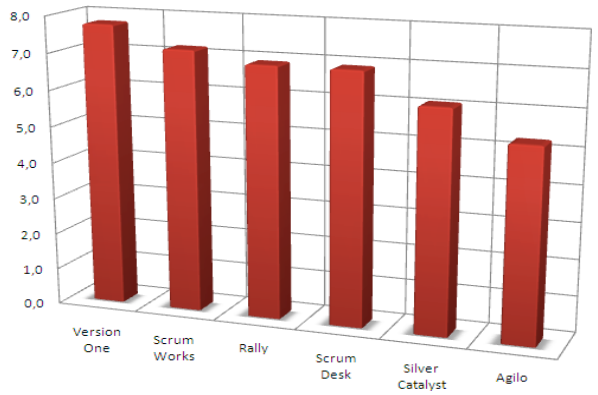

Figure 1. Summary chart of the total normalized rating U_N^t for each tool

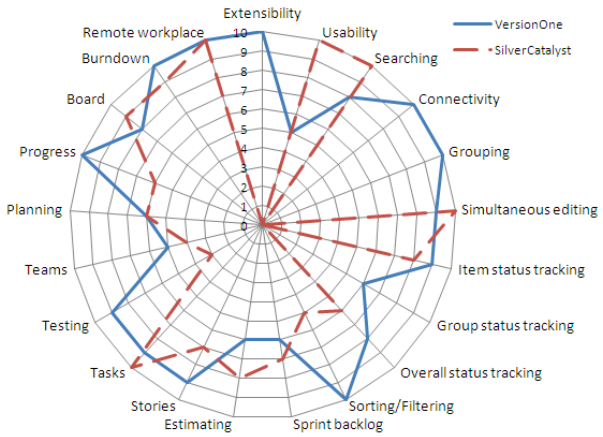

Figure 2. Comparison of the ratings of VersionOne and Silver Catalyst

TABLE III. RATINGS, TOTAL RATINGS, AND TOTAL NORMALIZED RATINGS FOR ALL CRITERIA AND ALL TOOLS

Criterion	Version One	Scrum Works	Rally	Scrum Desk	Silver Catalyst	Agilo
Extensibility	0,0	0,0	0,0	0,0	0,0	10,0
Usability	5,0	10,0	3,0	5,0	10,0	4,0
Searching	8,0	6,0	10,0	6,0	10,0	4,0
Connectivity	10,0	7,0	5,0	6,0	0,0	5,0
Grouping	10,0	9,0	7,0	0,0	0,0	4,0
Support for simultaneous editing	9,0	10,0	10,0	10,0	10,0	9,0
Story status tracking	9,0	10,0	9,0	9,0	8,0	4,0
Group status tracking	6,0	8,0	6,0	5,0	0,0	0,0
Overall status tracking	8,0	5,0	9,0	9,0	6,0	0,0
Sorting/Filtering	10,0	5,0	9,0	9,0	5,0	2,0
Sprint backlog	6,0	7,0	5,0	7,0	7,0	3,0
Estimating	6,0	5,0	8,0	9,0	8,0	6,0
Stories	9,0	10,0	7,0	7,0	7,0	5,0
Tasks	9,0	10,0	8,0	10,0	10,0	8,0
Testing	9,0	4,0	7,0	4,0	3,0	3,0
Teams	5,0	7,0	7,0	8,0	4,0	10,0
Planning	6,0	4,0	7,0	7,0	6,0	4,0
Progress	10,0	6,0	8,0	6,0	6,0	5,0
Board	8,0	10,0	0,0	9,0	9,0	6,0
Burndown	10,0	8,0	10,0	9,0	9,0	9,0
Remote workplace	10,0	10,0	10,0	10,0	10,0	10,0
Total rating	**163,0**	**151,0**	**145,0**	**145,0**	**128,0**	**111,0**
Normalized total rating	**7,8**	**7,2**	**6,9**	**6,9**	**6,1**	**5,3**

criterion (see Table 4), Scrum Master A assigned a weight of 0, while Scrum Master B assigned a weight of 10. The average weight for the *Progress* criterion is 5.3. This does not in any way reflect the company's overall need; neither does it reflect the individual needs. For instance, Scrum Master A saw no tool support need for tracking the team's progress since his team members were good at reporting the progress during stand up meetings every day, while Scrum Master B's team members were not good at reporting their progress and, therefore, it was necessary to track it via a tool. A similar conclusion can be made for the overall *Status Tracking* criterion.

It is interesting to compare the *Progress* criterion with the *Sorting/Filtering* criterion. The average ratings for these criteria were nearly the same – 5,3 and 5,7, respectively. However, unlike the greatly varying ratings assigned to the *Progress* criterion, all three Scrum Masters assigned weights of nearly the same value (5, 6, and 6) for *Sorting/Filtering*. Therefore, in this case we may draw a conclusion that the *Sorting/Filtering* criterion is of roughly the same importance to all Scrum Masters, and this is accurately reflected in the final average rating of 5,7. In contrast, the actual importance of the *Progress* criterion was lost after calculating its average value.

Calculating average weights for people of different roles, such as APMs and Scrum Masters, resulted in even less meaningful weights. Each role only saw the problems and requirements in their own area and daily work, and therefore, while rating they did not pay heed to the overall process and needs. In some cases, they made incorrect assumptions about the needs of others. For example, managers gave high ratings to virtual task boards, deeming it an important feature for the teams to have, while in reality the teams preferred to work with physical tools such as whiteboards. Thus, it became especially meaningless to calculate average ratings for different roles.

TABLE IV. WEIGHTS ASSIGNED BY THREE SCRUM MASTERS

Criteria	Scrum Master A	Scrum Master B	Scrum Master C	AVG
Extensibility	5	6	5	5,3
Usability	10	8	10	9,3
Connectivity	2	2	6	3,3
Searching	8	5	8	7,0
Grouping	10	10	8	9,3
Support for simultaneous editing	10	10	10	10,0
Story status tracking	10	4	7	7,0
Group status tracking	10	8	9	9,0
Overall status tracking	0	6	8	4,7
Sorting/Filtering	5	6	6	5,7
Sprint backlog	0	10	10	6,7
Estimating	10	10	9	9,7
Stories	0	6	8	4,7
Tasks	0	6	3	3,0
Testing	0	0	5	1,7
Teams	8	4	2	4,7
Planning	10	4	5	6,3
Progress	0	10	6	5,3
Board	0	4	4	2,7
Burndown	0	10	7	5,7
Remote workplace	0	6	8	4,7

The conclusion drawn from these results was that a similar evaluation method could not, and should not be used for determining the company's needs and for selecting tools.

VI. COMPANY OBSERVATION

In this section, we describe the results of our company observation. We first describe the status of the agile process at Company A. We then evaluate the process using the results of our interviews and our own observations.

VI.1 Status of the Agile Development Process at Company A

The overall process at Company A consisted of four main phases, (1) *Requirements definition*, (2) *Requirements management*, (3) *Project management*, and (4) *Development*. Figure 3 presents a simplified diagram of the company's process. We gathered an understanding of this process from the replies to interview *Question 1* presented in Table 2, as well as our own direct observations.

In the *Requirements definition* phase, new requirements were brought in from the customers in form of high-level specifications. They were all described according to a predetermined template and recorded in an MS PowerPoint file. For the sake of following the course of events and their related documents, we call this file the *Requirements Description File.* No specific reason was provided to why MS PowerPoint was being used. However, our impression was that the Program Managers, SPMs, APMs and Release APMs involved in the creation of requirements were accustomed to using MS PowerPoint. Once requirements had been created, they were then sent to the Release APM for approval

In the *Requirements management* phase, the new requirements were discussed, prioritized and decided upon. First, roles such as SPMs, Business Owners, APMs, UXDs, the Solution Architect, and the Release APM attended meetings during which the requirements were discussed and prioritized. To provide a basis for the meetings, the requirements from the *Requirements Description File* were put on a list that was stored in an MS Word document. We call this list *New Requirements List*. It contained a table with the requirement ID, a brief summary of some of the information taken from the *Requirements Description File*, and the requirement owner's name.

Once prioritization had been made, the prioritized requirements were manually added to a list of the existing requirements, which was stored in an MS Excel spreadsheet.

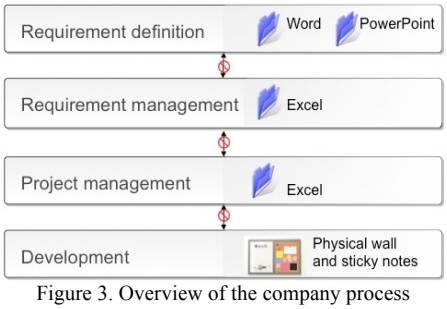

Figure 3. Overview of the company process

We call this list *Prioritized Requirements List*. It contained hundreds of requirements, along with a summary of information, such as, for instance, which requirements were defined for which customers. Still, the detailed descriptions of the requirements were stored in the *Requirement Description Files* in MS PowerPoint. Thus, at the end of this phase information about the same requirements was recorded in three different places: *Requirement Description Files, New Requirements List* and *Prioritized Requirements List*.

In the *Project management* phase, APMs and development teams broke down the requirements from the *Prioritized Requirements List* into lower-level, functional tasks. These tasks were then added to a list stored in another MS Excel spreadsheet, which we call *Product Backlog*. No links were provided to relate the backlog items to the high-level requirements, and vice versa. The backlog items were continuously updated, edited, and prioritized by different teams on different nodes. To make this possible, the *Product Backlog* was stored on a shared network drive, so that everyone on the internal company network could access it. At this point, requirements were described in four different places – the three files mentioned above and the *Product Backlog*.

Although the *Product Backlog* was accessible to everyone working on it, the company still encountered problems prioritizing the backlog items. In an ideal scenario, the prioritization should be made on a requirement level. In the company's scenario, however, this was not the case. Dependencies between different backlog items belonging to different requirements strongly affected the order in which the individual backlog items needed to be implemented. This complicated the process of prioritizing the requirements and monitoring their fulfillment. To add zest to it, some *Product Backlog* items belonged to two different requirements, creating many-to-many relationships.

In the *Development* phase, APMs together with the teams of different nodes performed Sprint planning. Here, they selected the highest prioritized items from the common *Product Backlog,* agreed on which node would implement them and included them in the team's respective *Sprint Backlog*s. The *Sprint Backlog* items were then recorded on a piece of paper affixed on a wall. They were further broken down into tasks and written on sticky notes. The information about the actual work that was done by the teams was not recorded anywhere else other than on the walls. At the end of the *Development* phase, the information about one requirement was stored in six different places, namely the four files that had been created at the end of the *Requirements management* phase, and two new places – the *Sprint Backlog* and sticky notes.

VI.2 Our Observations

The above-described scenario entailed a number of difficulties when managing and maintaining hundreds of requirements stored in different places. Using the feedback from *Question 3* in Table 2 (the question dealing with the pain areas of the company's process), we conclude that there were two main problems (1) *lack of visibility* and (2) *lack of traceability*.

Lack of visibility implied that different roles, especially the managerial ones, had no insight into the overall agile process. *Lack of traceability* implied that there was no link between the six artifacts storing information about the requirements. Both problems made it impossible to track the status of the requirements and to make sure that they had been completed.

The usage of different files for storing requirements created big difficulties. The files included hundreds of items, and it was difficult to detect similar requirements, group related requirements, as well as find their relationships, conflicts and duplicates. It was difficult to navigate among the files in order to get a complete overview of the requirements. In many cases, the contents in the files were not consistent. Changes made to one file were not always reflected in other files.

Many issues arose while using the *Product Backlog*, since it did not support simultaneous editing. Several nodes, teams, and APMs used the same *Product Backlog*, and they were thus unable to make any changes while somebody else was editing it. This naturally led to progress hindrance and frustration.

Using the answers to *Question 2* in Table 2 (dealing with the satisfaction with the current process), we discovered that the teams saw no problems with the process and its supporting tools. They were quite satisfied with it. They did not wish to change the process and to replace the physical walls with virtual task boards. This, however, was not the case with the mangers. The interviewed managers were not directly satisfied with the process and the tool supporting it.

Since the *Sprint Backlog* was only recorded on the walls, managers had no overview of the work progress in the teams. Moreover, they were usually too busy to attend the Sprint demos, which further meant that they were not always aware of what was completed and what was not completed. They did not always have good insight into the development process; they had no apprehension of team velocities, focus factors and other data. All this created difficulties in planning for future steps, managing resources, and dealing with customers.

Another difficulty faced by the managers was lack of access to release planning tools. Managers were forced to manually create release time plans by making drawings in MS PowerPoint and MS Word. To get an overview, they had to print them out on several sheets of paper and affix them on the walls. Due to the unavailability of a proper reporting tool, as well as lack of supporting data, they were also unable to create much needed reports providing various statistics on, for instance, number of incoming requirements and their rate of completion, task load on different nodes or teams, or number of requirements per customer. Finally, due

 ISBN: 978-1-61208-230-1

to lack of support for tracking changes to requirements, managers had to manually find out who made these changes, when, and why.

In general, we conclude that except for the implementation phase, all the process phases lacked appropriate tool support for the reasons described above. All this had led to a very cumbersome and clumsy management process. Using this as feedback along with the answers to *Questions 4* and *5* in Table 2, (questions eliciting needs for change), we extracted the company's six primary tool support needs. They are all displayed in Table 5 and motivated and matched to the selected tools in the section to come.

VII. COMPANY NEEDS AND TOOL FEATURES

In this section, we describe the company's needs listed during the *Company observation* phase and presented in Table 5. We then match each need against the features offered by the tools that had been evaluated during the *Tool evaluation* phase of our research.

Need 1: Support for management levels

The first need experienced by the company was *support for management levels*. The development teams were satisfied with the used physical tools. The managers, on the other hand, lacked tool support in the three management phases – *Requirements definition, Requirements management*, and *Project management.* They needed to store all their requirements in one centralized location. They needed a tool that would enable them to track the status of the requirements throughout all the management phases as well as to create various status reports. A similar need has been observed in [30].

In order to adequately support the above-described need, it would be necessary – though by no means sufficient – for a tool to provide support for hierarchical, multi-level requirements management. The tool should make it possible to store and track all the information that is stored in the *Requirements Description,* the *New Requirements List,* the *Prioritized Requirements List*, and the *Product Backlog*.

Looking at our tool evaluation from the point of view of this need, we saw that three out of the six tools studied provided the option of grouping *Product Backlog* items. One of the tools even included an artifact equivalent to the *New Requirements List*. However, none of the tools supported the hierarchical structure of the company's process. In fact, none of the evaluated tools supported high-level requirement definition, decision-making and prioritization in an agile process. Finally, none of the tools provided support for creating status reports.

TABLE V. SUMMARY OF THE COMPANY'S NEEDS

Company needs for Agile Tool Support	
1	Support for management levels
2	Simple and easy-to-use interface
3	Customized views for different roles
4	Support for the company's agile process
5	Adherence to company-specific terminology
6	Flexibility to adapt to process changes

The tools studied provided extensive support for the development level. All of them received a high rating in the provided options for breaking down *Product Backlog* items into tasks and for storing the tasks. They also provided detailed options for estimating the tasks and entering information regarding the work completed on the tasks. Finally, as many as five of the six evaluated tools provided virtual task boards. From the perspective of the company's need, however, these features were unnecessary and even undesirable, since the company wished to continue using physical walls for storing the *Sprint Backlog* and the team-level tasks. The company management had, however, a need to have an overview of the development phase in form of team velocities and completion of the *Product Backlog* items. This means that although there was no need to use virtual task boards, at least some information from the teams had to be channeled up for status tracking. Some of the evaluated tools had good support for status tracking, but none of them included artifacts for storing and handling the company's three levels of requirements. This made it impossible to use the tools to get a progress overview at the company.

Summing up, the tools studied focus on supporting team-level aspects of the agile process, such as virtual boards and support for Sprint planning. They do not provide enough coverage for the agile project management process that was desperately needed by the company.

Need 2: Simple and easy-to-use interface

The second need experienced by the company was *simple and easy to use interface* to be possessed by the tool. The tools used by the company were PowerPoint files, Word files, spreadsheets, slides, and sticky notes. All these are considered to be simple tools, and yet their usage became complicated because they did not adequately meet the company's needs. A similar need was reported in [31].

The company representatives expressed the need for "*just enough*" tool support, with simple, easy to use interfaces. This was valuable for the company since it had a complex structure, with multiple teams, roles, and process steps, and it was not desirable to introduce a tool that would further complicate daily work.

A requirement of simplicity implied that the tool had to be closely tailored to the company's process and it had to provide all the necessary features without providing the unnecessary ones. Our tool evaluation, however, revealed that the tools studied could not satisfy this need. They fell into two categories: (1) they were either simple and had pleasant interfaces, but did not offer advanced features, or (2) they were very complex and offered a wide array of options, but they were not easy to use. In general, we discovered that the inclusion of more features and options led to a decreased usability.

 ISBN: 978-1-61208-230-1

Need 3: Support for the company's agile process

The third need that the company experienced was *support for company's agile process*. The company had a complicated organizational structure and a multi-step requirements management process, where numerous people of different roles were involved.

The company wished to keep and have power over their process. It is highly undesirable for a company to be forced to introduce changes to their process in order to adopt a particular tool. Instead, the tool should be adapted to the company structure, the product it manages and the team setup. A tool should enable a workflow similar to that of the company, as well as store and display the information that is relevant to the company.

None of the tools satisfied this need. The tools studied were either simple and lightweight, or powerful and complex. The powerful and complex tools, such as *Rally*, imposed process adaptations. Even the simpler tools, such as *Silver Catalyst,* imposed their own process.

Need 4: Customized views for different roles

All roles were using the same spreadsheets and slides while working with requirements at the company. All the useful information had to be displayed in the same place. This created an overload of information for all the roles involved. For example, a UX designer did not need to see the same information as an APM performing a breakdown of *Product Backlog* items, while SPMs were mostly interested in looking at reports and charts.

The company needed to have different views for different roles in order to make their daily work simpler and more manageable. Hence, the fourth need identified concerned *customized views for different roles*.

Our tool evaluation revealed that the more advanced tools did account for a few different roles, but the views they provided and the information they displayed were not sufficient. The existing agile project management tools have failed to predict the need for all the views and custom reports that a company might need, especially in case of a large company with a large number of different roles.

Need 5: Adherence to company-specific terminology

The terms, concepts and abbreviations used at the company were quite complex and differed from the terminology used outside the company. For example, the company used the term "opportunity card" instead of "high-level requirement." The term was used by a large number of people and in a large number of documents. Changing this term to fit a particular tool was not an option. The company needed a tool that made it possible to adhere to the terminology already in place.

Not surprisingly, the tools we evaluated imposed their own terminology. Hence, they did not fulfill this need. For example, some tools used the concept of a "feature" to describe groups of *Product Backlog* items. In the company, however, features were certain groups of functionalities, and the term did not coincide with the way it was used in agile project management tools.

Need 6: Flexibility to adapt to process changes

Changes in the process models and the ways of working were not an infrequent occurrence at the company. Adhering to the agile vision of continuous improvement, there was a drive to continuously learn from retrospectives and make process improvements. The company needed a tool that would not only support their current process, but would also accommodate process changes and an evolving company structure. Thus, the company's sixth need was *flexibility to adapt to process changes*.

Five out of six evaluated tools were of commercial availability and could not be extended to add or remove desired features. They simply did not support an evolving company process. Hence, they did not fulfill this need.

VIII. CONCLUSION

Despite the growing availability and complexity of agile tools, there is lack of tool selection guidelines and case studies. In this paper, we have attempted to shed some light on the matter by presenting the results of a case study of agile tool selection conducted at Company A – a company with a complex process and hierarchical requirements structure. As part of our research, we performed an evaluation of six agile tools available on the market using a detailed list of evaluation criteria.

During our attempt to select an adequate tool for Company A, we found out that the tool evaluation criteria, though well defined, proved to be insufficient for specifying the company's needs. We, however, do not reject evaluations of this type as an important aid in identifying needs. They have to be complemented with extensive observational studies and detailed interviews, similar to the study reported in this paper.

Even after having extracted and identified the company's needs, we had difficulties in finding an adequate tool. The tools focused more on team-level aspects of development and did not cater to the multi-layer requirements management process of the company. The tools available on the market imposed their own process and were cumbersome and difficult to use. They were not flexible enough to accommodate changes in the company's process, and they lacked support for the creation of reports. Further, the tools did not cater to the needs of all the roles present at the company, and, in most cases, imposed their own terminology. Overall, we conclude that the studied tools have not met Company A's agile project management needs.

It might be expected that other large companies with complex structures also face a similar dilemma. As future work, we plan to look into custom tools, or a combination of custom tools and tools from the market, in order to find out whether they might successfully meet the needs of such companies. We also plan to find out how current tools support distributed, multi-cultural agile environments that currently encounter many types of different problems [32].

 ISBN: 978-1-61208-230-1

REFERENCES

[1] M. Dubakov and P. Stevens, "Agile tools. The good, the bad and the ugly," TargetProcess Inc., 2008.

[2] G. Dowst. *Reviewing agile process management tools. Part 1 and 2* [Online]. Available: http://consultingblogs.emc.com/gavyndowst [retrieved: October, 2012].

[3] B. Swanson (2009 Sep. 25). *Comparing open source agile project management tools.* [Online]. Available: http://olex.openlogic.com/wazi/2009/comparing-open-source-agile-project-management-tools [retrieved: October, 2012].

[4] CMC Media Inc. (2007 Apr.). *Agile Tooling: A Point, Counter-Point Discussion* [Online]. Available: http://agile.techwell.com/articles/weekly/agile-tooling-point-counter-point-discussion [retrieved: October, 2012].

[5] G. Goth, "agile tool market growing with the philosophy," IEEE Software, 2009, pp. 88-91.

[6] Mountain Goat Software. *All products*. [Online]. Available: http://userstories.com/products [retrieved: October, 2012].

[7] S. H. Rayhan and N. Haque, "Incremental adoption of Scrum for successful delivery of an IT project in a remote setup," in Proc. AGILE 2008 Conference, IEEE Computer Society, Toronto, Canada, 4-8 August 2008, pp. 351-355.

[8] M. Cottmeyer, "The goods and bad of Agile offshore development," in Proc. AGILE 2008 Conference, IEEE Computer Society, Toronto, Canada, 4-8 August 2008, pp. 362-367.

[9] E. Uy and R. Rosendahl , "Migrating from SharePoint to a better Scrum tool" in Proc. AGILE 2008 Conference, IEEE Computer Society, Toronto, Canada, 4-8 August 2008, pp. 506-512.

[10] F. Cannizzo, G. Marcionetti, and P. Moser, "Evolution of the tools and practices of a large distributed Agile team" in Proc. AGILE 2008 Conference, IEEE Computer Society, Toronto, Canada, 4-8 August 2008, pp. 513-518.

[11] VersionOne Inc.. *State of Agile Development Survey 2009* [Online]. Available: http://pm.versionone.com/StateOfagileSurvey.html [retrieved: October, 2012].

[12] P. Behrens, "agile Project Management (APM) tooling survey results," Trail Ridge consulting, December 2006.

[13] VersionOne Inc. [Online]. Available: http://www.versionone.com [retrieved: October, 2012].

[14] CollabNet Inc. *ScrumWorks.* [Online]. Available: http://www.open.collab.net/products/scrumworks/?q=scrumworks [retrieved: October, 2012].

[15] Rally Software Development Corp. *AgileZen* [Online]. Available: http://agilezen.com [retrieved: October, 2012].

[16] ScrumDesk Co. [Online]. Available: http://www.scrumdesk.com [retrieved: October, 2012].

[17] Silver Stripe Software Pvt. , Ltd.. *Silver Catalyst* [Online]. Available: http://toolsforagile.com/silvercatalyst [retrieved: October, 2012].

[18] Agile42 Gmbh. *Agilo* [Online]. Available: http://www.agile42.com/cms/pages/agilo [retrieved: October, 2012].

[19] VersionOne Inc. *Integrations* [Online]. Available: http://community.versionone.com/sdk/Documentation/Integrations.aspx [retrieved: October, 2012].

[20] Bugzilla org. [Online]. Available: http://www.bugzilla.org/ [retrieved: October, 2012].

[21] Atlassian Pty Ltd. *JIRA issue and project tracking* [Online]. Available: http://www.atlassian.com/software/jira [retrieved: October, 2012].

[22] Eclipse org. [Online]. Available: http://www.eclipse.org [retrieved: October, 2012].

[23] Microsoft Inc. *MS Excel*. [Online]. Available: http://office.microsoft.com/en-us/excel [retrieved: October, 2012].

[24] Rally Software Development Corp. *Rally Connectors* [Online]. Available: http://www.rallydev.com/agile_products/integrations/connectors/ [retrieved: October, 2012].

[25] Microsoft Inc. *Visual Studio Team Foundation Server* [Online]. Available: http://msdn.microsoft.com/en-us/vstudio/ff637362.aspx [retrieved: October, 2012].

[26] Mantis Bug Tracker. [Online]. Available: http://www.mantisbt.org [retrieved: October, 2012].

[27] Subversion SVN [Online]. Available: http://subversion.apache.org [retrieved: October, 2012].

[28] Trac [Online]. Available: http://trac.edgewall.org [retrieved: October, 2012].

[29] Wikipedia. *User experience design* [Online]. Available: http://en.wikipedia.org/wiki/User_experience_design [retrieved: October, 2012].

[30] M. Kajko-Mattsson. "Problems in agile trenches". In Proc. of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '08). ACM, New York, NY, USA, 2008, pp. 111-119.

[31] G. Azizyan, M. K. Magarian, and M. Kajko-Matsso "Survey of Agile Tools Usage and Needs" in Proc. AGILE 2011 Conference, IEEE Computer Society, Salt Lake City, UT, USA, 8-12 August 2011, pp. 29-38.

[32] M. Kajko-Mattson, G. Azizyan, and M. K. Magarian "Classes of distributed Agile problems" AGILE 2010 Conference, IEEE Computer Society, Orlando FL, USA, 9-13 August 2010, pp. 51-58.

 ISBN: 978-1-61208-230-1