Describing Criteria for Selecting a Scrum Tool Using the Technology Acceptance Model

Describing Criteria for Selecting a Scrum Tool Using the Technology Acceptance Model

Gerard Wagenaar[1(✉)], Sietse Overbeek[2], and Remko Helms[3]

[1] Avans University of Applied Sciences, Breda, The Netherlands
g.wagenaar@avans.nl
[2] Utrecht University, Utrecht, The Netherlands
s.j.overbeek@uu.nl
[3] Open University, Heerlen, The Netherlands
remko.helms@ou.nl

Abstract. Scrum teams extensively use tools to support their processes, but little attention has been given to criteria a Scrum team applies in its selection of such a tool. A greenfield approach was used to explore these criteria. To this extent twelve Scrum teams were asked to list criteria and assigned weights in their decision processes. After having chosen and used a tool for a number of Sprints, the teams also evaluated the selected tools. Using the Technology Acceptance Model to structure findings, two major categories were identified: Perceived usefulness, alias criteria directly related to Scrum, and perceived ease of use. Most teams listed more or less the same criteria. Within the categories several specific subcategories were distinguished, for instance burn-down chart support or multi-platform aspects. Teams evaluated more issues, positive or negative, within the Scrum-related criteria. The findings indicate that Scrum teams prefer perceived usefulness over perceived ease of use. In other words: Specific support of Scrum, especially its artefacts, are of greater value to a team than general tool considerations.

Keywords: Agile · Scrum · Supporting tool · Technology Acceptance Model (TAM) · Weighted criteria

1 Introduction

Agile software development (ASD) prefers "*Individuals and interactions over processes and tools*" with an annotation: *"... there is value in the items on the right ..."* [1]. Tools have proven to be valuable, as, for instance, global Scrum projects depend on a wide range of tool support [2]. This includes communication and collaborative tools for issue or bug tracking, backlog management, and burn-down chart visualization. Also in a collocated project a tool proved to effectively manage a Scrum project [3]. In general, communication and collaboration tools, whether used face-to-face or global, are crucial in modern agile teams [4].

Agile tools come in abundance [5]; a choice has to be made by an agile team. However, knowledge of criteria applied by a team in this choice is anecdotal and

N.T. Nguyen et al. (Eds.): ACIIDS 2017, Part II, LNAI 10192, pp. 811–821, 2017.
DOI: 10.1007/978-3-319-54430-4_77

fragmented, whereas such knowledge gives insight in where agile teams need support. This in turn deepens our theoretical understanding of agile processes through the explicit articulation of important ASD elements by its practitioners, while at the same time practically supporting them in choosing appropriate support. Because of our approach, a greenfield one, our results would especially be useful for novel teams involved in an agile transformation.

The remainder of this paper is organised as follows. In Sect. 2 we discuss related work; in Sect. 3 we present our research method. Results are shown in Sect. 4, followed by a discussion and conclusions in Sect. 5, which is the final section.

2 Related Work

Research on criteria agile teams use for tool support has not been extensive so far. Fragmented research has been presented, which can be captured under two headings: (1) Individual experiences [6–8] and (2) Research explicitly addressing the need for a more general approach [9, 10]. However, all tend to assess tools on the basis of existing criteria rather than to tackle the problem of first defining criteria for the selection of tools thereafter. We now first discuss both categories and introduce a more general theoretical viewpoint afterwards.

2.1 Experience Reports

Experience reports describe a selection process or, somewhat more general, a transition process towards ASD. Uy and Rosendahl [6] discuss a case where a technology team, responsible for a corporate website as well as a number of other products and information systems, entered a migration trajectory from SharePoint to a tool better suited for Scrum. Three critical factors were used to distinguish between two short-listed alternatives (Rally, VersionOne): Usability, functionality, and configurability. Additional technical considerations concerned were in the areas of enterprise infrastructure, architecture and quality assurance.

Møller et al. [7] designed a Scrum tool dedicated to assist in a daily Scrum meeting with four overall requirements: Intuitive user interface, high accessibility, commitment to Scrum, and project history. The requirements were found on the basis of experiences of three Scrum teams.

Engum et al. [8] report on a small North European agile company, which transitioned to a tool for managing Product and Sprint Backlog and distilled lessons from its experiences. Impediments the tool should address included task description and tracking, specifically in order to help organize, specify and prioritize the product backlog and track development, and time usage.

Whether phrased as guidance [8], tips [3] or lessons learned [6] all experience reports acknowledge the fact the criteria used are situational, that is, driven by the specific circumstances of the organization, the project, or the team. This is no surprise, since it is already known that a software development process itself is already regarded as being dependent on the situational characteristics of individual software

development settings [11]. If the choice of a process itself is already situational, the choice of a tool to support the process is situational too.

2.2 Criteria Models

To move beyond the experiences from a single case, more recent research explicitly addresses the need for this general approach, especially modelling criteria for tool evaluation. Azizyan et al. [9] investigated (dis)satisfactory aspects of agile tools through a survey of agile tool usage and needs. The criteria used to evaluate agile tools were: Ease of use, integration with other systems, customizability, availability of reports, and price.

Taheri and Sadjad [10] used a comparison chart with criteria in a selection process for a tool: Lifecycle coverage, Simplicity & ease of use, Collaboration, analytics, visibility & reporting, workspace & process, program management, deployment, and integrity & security. The criteria originated, beside from previous research, for a large part from surveys, sponsored by vendors of agile tools. This introduces a bias, since vendors are of course unlikely to use unfavourable criteria.

2.3 Theoretical Viewpoint

Both criteria models, and also the individual experiences, draw heavily upon existing surveys, where their resources are subject to bias. To have a more independent viewpoint the Technology Assessment Model, TAM [12, 13] can be used. Its central proposition is that user acceptance of technology depends on perceived usefulness and perceived ease of use. Perceived usefulness is "*the degree to which a person believes that using a particular system would enhance his or her job performance*"; perceived ease of use "*the degree to which a person believes that using a particular system would be free of effort*" [12, p. 320]. TAM theorizes that the effects of external variables (e.g., system characteristics, development process, and training) on intention to use are mediated by perceived usefulness and perceived ease of use [14]. Both categories are also included in the Unified Theory of Acceptance and Use of Technology (UTAUT) as performance and effort expectancy, as are 2 others: social influence and facilitating condition [15]. TAM thus provides a useful first distinction in modelling criteria for agile tool support. This is confirmed, although not explicitly by the use of TAM, in an evaluation of four agile project management tools, where both usability evaluation criteria and a task-oriented usability inspection were used [16]; these categories reflect very well the TAM categories. This supports the use of dichotomy in TAM in modelling criteria that agile teams use for tool support.

3 Research Method

In our research we have chosen for a greenfield approach, in analogy with the greenfield project: "*In software engineering jargon, a greenfield is a project which lacks any constraints imposed by prior work*" [17, p. 21]. Although this approach may

be considered as situational as any other, it provides a relatively unbiased way to explore features a Scrum team thinks of as being important in its support. It at least assures that prior organizational or commercial influences are excluded. To implement this approach we used student groups, where we phrased our research question as follows:

Which criteria do Scrum teams adopt when choosing computer-based support for their Scrum process?

We characterize our research as a variant of case study research with the teams being the cases, but explicitly instructed through their assignments.

In the next two paragraphs we will describe, first, the educational context of the student groups and, second, the assignments the groups were provided with.

3.1 Context

The teams in this study are teams of second year students in a four year Information Technology study programme leading to a Bachelor degree. Students have experience with software development methods, some basic experience with waterfall-like development, followed by other methods, such as RUP (Rational Unified Process). In ten weeks students were to develop an application, applying Scrum as ASD method.

In the first three weeks of the course students acquainted themselves with Scrum. After a lecture on agile methods in general students were referred to the Scrum guide [18] for self-study and conducted a test consisting of twenty multiple choice questions afterwards. Furthermore, they were provided with training on all major Scrum elements, which are roles, artefacts and events.

From week four onwards the students made three Sprints of four days each, working fulltime on their application. During a Sprint teams held all Scrum meetings: Sprint Planning Meeting (Monday morning), Daily Scrum (Tuesday till Thursday morning), Sprint Review, followed by an informal Sprint Retrospective (Thursday afternoon). Teams were stimulated to collectively work at the same place, for instance a computer or a conference room. A few teams had their members working (more) individually, sometimes from home. From week seven onwards students continued to work on the application individually using team results as a basis. This had didactical reasons and although still working in a Scrum-like way, this part of the course is left out of consideration here, if only because communication between team members did not take place anymore.

3.2 Assignments

Students were divided into twelve Scrum teams, with each team between four and six members. Before starting their first Sprint, teams were asked to select a tool to support them in their Scrum process as a first assignment. In more detail teams were asked to specify at least five weighted criteria they thought relevant for their Scrum tool, three different alternatives (Scrum tools), and a score on each criterion for each alternative. In addition, teams were asked to motivate all of the alternatives, criteria and scores.

As a second assignment, teams were asked for an evaluation after their last Sprint. They were specifically asked to answer two questions: On which characteristics did the tool (not) meet your expectations? In other words, where did the tool (not) effectively and efficiently support the Scrum process?

3.3 Data Collection and Analysis

For data collection, we first collected reports with regard to the first assignment from all twelve groups at once. We analysed the reports according to Grounded Theory (GT). GT is a systematic research method known for the generation of theory derived from systematic and rigorous analysis of data [19, 20]. This does not mean that there is not a specific problem, but rather problems and their key concerns emerge during data analysis [20].

To analyse the data, we used three abstraction levels: category, concept, code [21]. For the highest level, category, we adopted TAM: Perceived usefulness and perceived ease of use. Applying open and axial coding [22] we found at the next level concepts like support of a product or sprint backlog (perceived usefulness) or multi-platform support (perceived ease of use). Through coding different wordings were mapped onto the same (sub-)criteria. For instance, the criterion "Ease of use" was phrased as: "*The users must be able to use it easily*", "*The cognitive load should be low*", and "*Often used functions should be clearly visible*". Coding was a fairly straightforward exercise, because the assignment by its nature already led to structured reporting.

Our analysis proceeded from one report to another; we coded a report for criteria and weights, adding to our abstraction levels where necessary. After each change we scrutinized previous reports with respect to the most recent supplements. This was for instance necessary when we met a criterion previously considered as sub-criterion, or vice versa. This overlap between criterion and sub-criterion happened more often and has been resolved by using our judgement in the analysis, supported by the classification already applied by the teams themselves and the number of teams that used the one or the other.

We also collected reports for the second assignment. We analysed them for positive and negative remarks, but only superficially coded them, because most teams did not explicitly refer to their (sub-)criteria, thus allowing coding at the category and concept level only.

4 Results

The results from our research are described next. We first describe the criteria which were used by the teams to evaluate the tools and show the weights teams attached to them. We then provide a summary of the tools considered by the teams and present a brief overview of the teams' evaluation after having used their tool.

4.1 Criteria

All twelve Scrum teams reported criteria, alternatives, scores as well as considerations leading to them. They reported their sources as (Internet) research, acquaintance with a tool (for instance through experience from a part time job), advice from experts, including teaching staff, or a combination. Since the teams worked independently, we applied coding to highlight similarities and discrepancies, especially between (sub-) criteria, which includes homonyms and synonyms.

Six teams used a flat list, that is, all criteria were on the same level of abstraction. The others had a hierarchical approach, in which a classification of criteria with sub-criteria was used, yielding a two level hierarchy. After analysis a hierarchical list of criteria and sub-criteria arose, structured at the top level according to TAM (Table 1);

Table 1. Overview of categories, criteria and sub-criteria

Category Criterion *Sub-criterion*	#[1]
Perceived usefulness	1
Contains visualization task board	5
split in to do, doing, testing & done	*7*
Supports backlog(s)	-
contain product backlog (items)	*5*
contain sprint backlog (items)	*5*
assign priorities to items	*2*
split items in tasks	*3*
link items/tasks to team members	*4*
allow comments on item/task	*5*
allow documents per item/task	*2*
register time per task	*1*
sort backlog items on priority	*2*
show status backlog item	*1*
show activities per member	*1*
show deadlines for tasks	*3*
show effort (per member)	*1*
show statistics	*4*
couple sprint backlog to releases	*1*
Supports burn-down chart	8
show team	*1*
show individual	*1*
Has additional features	1
Has Scrum compatibility	2
allow (user) stories	*1*
have predictions	*1*
support planning poker	*1*

Category Criterion *Sub-criterion*	#
Perceived ease of use	
Ease of use	9
can be personalized	*3*
have an easily accessible lay out	*3*
have short loading & response time	*2*
have user guide / tutorial	*1*
Multiple platform support	7
have a website	*1*
be app/application/board	*4*
be physical	*1*
can be reached from other locations	*1*
Communication support (e-mail, chat)	2
Security mechanism support	-
require log in	*2*
administer member rights	*3*
Integration with other tools	-
Visual Studio	*1*
GIT	*2*
Acceptable price (or trial available)	10

[1] # is the number of occurrences of the (sub-)criterion.

this list is the superset of all individual lists. All (sub-)criteria could be classified as a TAM category, perhaps with the exception of price, which we included under perceived ease of use, but might have had its own category.

The frequency for criteria, not being sub-criteria, represents the number of times the criterion itself is mentioned directly, i.e. it does not in any way accumulate frequencies of sub-criteria.

All criteria and sub-criteria were rated with relative importance by the teams, see Table 2.

Table 2. Overview of weights of criteria

Category	Criterion	Teams	Weight[2]	σ[3]
Perceived usefulness	Contains visual task board	8	12,5	14
	Supports backlog(s)	10	27,8	19
	Supports burn-down chart	9	8,3	6
	Has additional features	1	1,3	4
	Has Scrum compatibility	3	3,4	6
Perceived ease of use	Ease of use	10	17,2	16
	Multiple platform support	11	11,8	9
	Communication support (e-mail, chat)	2	2,1	5
	Security mechanism support	3	2,3	5
	Integration with other tools	3	3,3	7
	Acceptable price (or trial available)	10	10,0	7

[2]All numbers are rounded to 1 decimal.
[3]σ: Standard deviation, rounded to an integer.

In Table 2 the number of teams refers to teams that have listed the criterion or one of its sub-criteria. The weight is calculated as the sum of the relative weights for teams who listed the criterion and/or one of its sub-criteria divided by the total number of teams (twelve). To this extent the scale of weights for each individual team has been normalized to a scale of 0–100. Furthermore, we have restricted ourselves to the level of criteria in Table 2. Incorporating sub-criteria in the table would have little value, because the vast majority of sub-criteria is mentioned three times or less, which would in general result in lots of weights smaller than one.

4.2 Choice of Tools and Evaluation

Teams were free in their choice of alternatives. A total of eighteen different Scrum tools was considered. All tools were fairly dedicated Scrum tools with the exception of Trello (a tool to organize anything), and Proofhub and Dapulse (project management tools). Five of them were finally chosen by the teams. Visual Studio Online (Team Foundation Server) was the most chosen one, five times, followed by Trello, four times. The other three teams used Scrumwise, ScrumDo and QuickScrum.

In the evaluation almost all teams reported to be satisfied with their choice of a tool. Their satisfaction was backed up by both positive and negative remarks of (the use of) their tools. We counted the number of remarks, related them to our two (TAM-) categories and subdivided them into positive or negative feedback (Table 3).

Table 3. Feedback of chosen tools

		Team												Total
		1	2	3	4	5	6	7	8	9	10	11	12	
Perceived usefulness	Positive feedback	7	2	4	1	1	1	4	3	2	3	1	5	34
	Negative feedback	3	3	2	3	–	–	1	1	2	1	1	–	17
Perceived ease of use	Positive feedback	2	1	–	4	5	5	3	–	1	1	2	–	24
	Negative feedback	2	–	–	1	–	–	–	1	–	1	1	2	8

We have related remarks to the category only, because the evaluation data most often was not related to only one of the (sub-)criteria. Some examples of remarks are:

- "*The basic features of the tool, such as the task board, the description of Sprint Backlog items and the planning of Sprints, were easy to use*".
- "*We would have liked planning poker points instead of hours to measure velocity*".

One team switched from tool after Sprint one. It had chosen for Trello, but started using Visual Studio Online for Sprint two and three.

5 Discussion and Conclusions

In this section we first compare our results with previous research. We then discuss the (sub-)criteria in detail and analyse the teams' preferences. Finally, we discuss limitations of our study.

5.1 Comparison with Previous Models

The teams collectively came up with 45 criteria, including sub-criteria, far beyond and more detailed than any of the previous models [9, 10]. However, our sheer amount still does not warrant a claim to completeness. And we do notice a match between criteria from previous research and our current research, at least at the level of criteria. Furthermore, there was consensus among the teams. Six criteria were mentioned by at least 75% of the teams (visual task board, backlog(s), burn-down chart, ease of use, multiple platform, acceptable price), with the remaining ones all below 25% (see Table 1).

We conclude that our Scrum teams, most likely unaware, followed theory (TAM) and practice (experience reports and models) in the constitution of their list of relevant (sub-)criteria, but added lots of detail concerning practical considerations, such as wishes to prioritize and sort backlog items or to integrate with GIT. This level of

detail was not established in previous research and in this way our results are especially attractive for teams in agile transformation, when adopting agile tools.

5.2 Classification of Criteria

We found criteria could easily be classified according to the TAM, with no overlap between Perceived usefulness and Perceived ease of use. Notwithstanding the high standard deviations in the attribution of weights (Table 2), a separation between '(sub-) criterion mentioned often/high weight' and '(sub-)criterion mentioned seldom/low weight' is clear and cuts through the categories: Teams value support of Scrum artefacts most, as is shown by the number of criteria devoted to the Product and Sprint Backlog (and the burn-down chart), and acknowledged by the weights they attached to those criteria. This observation is also confirmed by the evaluation where teams devoted more remarks, positive or negative, to Perceived usefulness than Perceived ease of use, approximately in a ratio 5:3 (Table 3). This suggests Scrum teams value perceived usefulness over perceived ease of use.

However, this is not confirmed in the scores they assign to their alternatives/criteria, and hence the choice of their tool. Trello is the second best chosen tool, and Trello may certainly be suited for the task, but it is a general tool and certainly not a Scrum specific one. However, the one team that switched from Trello to Visual Studio Online (VSO), when comparing the two, commented: "*VSO seems, as far as burndown chart and overall look & feel are concerned, far more professional than Trello*". "*Professional*" here also implies the existence of more and better Scrum features in VSO. Probably Trello's ease of use and flexibility lured some teams away from more specific Scrum tools.

We conclude that Scrum teams prefer perceived usefulness over perceived ease of use. Especially support of Scrum's artefacts is of greater value to a team than general tool considerations.

5.3 Limitations

We excluded commercial and organizational influences by involving student Scrum teams with basic and theoretical knowledge of Scrum. A drawback of this greenfield approach is that our findings apply to novice Scrum practitioners. Generalizability of results is therefore limited.

The teams tended to prefer freely available Scrum tools, or a trial version thereof. Although this may have restricted their choice of a tool, it did not influence the constitution of the list with criteria, which was our main research goal.

Acknowledgements. This research would not have been possible without the efforts of students taking the course. The first author also wants to express his gratitude to Avans University of Applied Sciences for facilitating and supporting this research.

References

1. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al.: Agile Manifesto (2001). agilemanifesto.org. Accessed 24 Sept 2012
2. Hossain, E., Babar, M.A.: Risk identification and mitigation processes for using Scrum in global software development: a conceptual framework. In: Proceedings of the Asia-Pacific Software Engineering Conference, Penang, Malaysia, pp. 457–464. IEEE (2009)
3. Schatz, B., Abdelshafi, I.: Primavera gets agile: a successful transition to agile development. IEEE Softw. **22**(3), 36–42 (2005)
4. Crowder, J.A., Friess, S.: Productivity tools for the modern team. In: Crowder, J.A., Friess, S. (eds.) Agile Project Management: Managing for Success, pp. 43–48. Springer, Heidelberg (2015)
5. Alyahya, S., Alqahtani, M., Maddeh, M.: Evaluation and improvements for agile planning tools. In: Proceedings of the International Conference on Software Engineering Research, Management and Applications, Baltimore, USA, pp. 217–224. IEEE (2016)
6. Uy, E., Rosendahl, R.: Migrating from SharePoint to a better Scrum tool. In: Proceedings of the Agile Conference, Toronto, Canada, pp. 506–512. IEEE (2008)
7. Møller, L.S., Nyboe, F.B., Jørgensen, T.B., Broe, J.J.: A Scrum tool for improving project management. In: Wouters, I., Man, Fl. K., Tieben, R., Offermans, S., Nagtzaam, H. (eds.) Flirting with the Future: Prototypes Visions By The Next Generation - Proceedings of the Student Interaction Design Research Conference, Eindhoven, The Netherlands, pp. 30–32 (2009)
8. Engum, E.A., Racheva, Z., Daneva, M.: Sprint planning with a digital aid tool: lessons learnt. In: Proceedings of the Conference on Software Engineering and Advanced Applications, Patras, Greece, pp. 259–262. IEEE (2009)
9. Azizyan, G., Magarian, M.K., Kajko-Mattson, M.: Survey of agile tool usage and needs. In: Proceedings of the Agile Conference, Salt Lake City, USA, pp. 29–38. IEEE (2011)
10. Taheri, M., Sadjad, S.M.: A feature-based tool-selection classification for agile software development. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering, Pittsburgh, USA, pp. 700–704 (2015)
11. Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: towards a comprehensive reference framework. Inf. Softw. Tech. **54**(5), 433–447 (2012)
12. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. **13**(2), 319–340 (1989)
13. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: a comparison of two theoretical models. Manag. Sci. **35**(8), 982–1003 (1989)
14. Venkatesh, V., Davis, F.D.: A theoretical extension of the technology acceptance model: four longitudinal field studies. Manag. Sci. **46**(2), 186–204 (2000)
15. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: toward a unified view. MIS Q. **27**(3), 425–478 (2003)
16. Alomar, N., Almobarak, N., Alkoblan, S., Alhozaimy, S., Alharbi, S.: Usability engineering of agile software project management tools. In: Marcus, A. (ed.) DUXU 2016. LNCS, vol. 9746, pp. 197–208. Springer, Heidelberg (2016). doi:10.1007/978-3-319-40409-7_20
17. Gupta, R.M.: Project Management. PHI Learning Pvt. Ltd., Delhi (2011)
18. Schwaber, K., Sutherland, J.: The Scrum guide – The Definitive Guide to Scrum: The Rules of the Game (2013). www.scrum.org. Accessed 17 Sept 2013
19. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, Chicago (1967)

20. Glaser, B.G.: Basics of Grounded Theory Analysis: Emergence vs Forcing. Sociology Press, Mill Valley (1992)
21. Allan, G.: The use of grounded theory as a research method: warts and all. In: Brown, A. (ed.) Proceedings of the European Conference on Research Methodology for Business and Management Studies, pp. 9–19. Management Centre International Ltd. (2003)
22. Corbin, J., Strauss, A.: Basics of Qualitative Research - Techniques and Procedures for Developing Grounded Theory. Sage Publications Inc., Thousand Oaks (2008)