

# Usability Engineering of Agile Software Project Management Tools

Noura Alomar<sup>(✉)</sup>, Nouf Almobarak, Sarah Alkoblan,  
Sarah Alhozaimy, and Shahad Alharbi

Software Engineering Department, College of Computer and Information  
Sciences, King Saud University, Riyadh, Saudi Arabia  
{n.alomar, n.almobarak, s.alkoblan,  
s.alhozaimy, s.alharbi}@ksu.edu.sa

**Abstract.** The successful management of software projects requires taking human and managerial factors into consideration. Agile software project management methodologies have made their way into the mainstream culture of software development and have gotten the attention of software engineers and researchers due to their rapid growth. The aim of this research effort is to comprehensively evaluate the usability of four software project management tools based on experimental findings as well as heuristic assessment. We focus on evaluating widely known tools based on rigorous usability assessment criteria and subjective and objective evaluation techniques. By utilizing the capabilities of a usability testing software solution, Morea, and considering the subjective views of five Human Computer Interaction experts, we believe that our findings can inspire the design of more effective agile software project management tools that allow development teams to manage their work efficiently while helping decision makers to base their tool selection on a trusted usability evaluation approach that addresses the needs of software development teams. We also believe that our findings will have promising implications for task management activities performed throughout all the phases of the software development lifecycle.

**Keywords:** Agile · Usability · Software engineering · Software project management · User experience

## 1 Introduction

With the growing demand for streamlining software project management activities by increasing the level of collaboration between individuals who work in software development teams, many agile software project management tools have been introduced. These software solutions offer a variety of features that mainly help software engineers distribute and prioritize development tasks and manage the overall progress of their projects. When deciding which software project management tool would better fulfill the needs of a given organization and help in achieving its business goals, looking for a complete solution that gives each member the ability to efficiently plan, track, and manage each iteration of the project is of extreme importance. Usability focuses on how end users will work with the software, and agile development processes focus on how software developers can flexibly deliver their assigned tasks.

Once a specific software project management tool is chosen by a team, the software analysts, designers, developers and testers are supposed to use the selected tool to track the progress of other members and collaborate to achieve the objectives of their project and thus contribute to satisfying their business goals. Therefore, although team members are assumed to have the technical expertise to use these software solutions, agile software project management tools that suffer from usability shortcomings might discourage team members from using the chosen tools and thus negatively affect the progress of their teams. For globally distributed teams, the lack of transparent and visible progress of all team members might lead to these teams' resources being wasted. For instance, some members might start working on work items that were already completed by other members. This in turn might complicate the integration of the development work done by distributed teams and increase inconsistencies between the work items completed by different teams, leading to cost and schedule overruns.

In this study, practical research is conducted on widely adopted tools and their features that are widely demanded in the software industry are explored. These tools are JIRA, AgileZen, VersionOne and ZebraPlan. The tool selection process was based on the support of the essential features that agile software development teams demand in practice, such as visibility of progress, communication, task prioritization and time management features. Before evaluating the usability of the chosen tools, we define comprehensive evaluation criteria covering elements chosen specifically to evaluate our participants' confidence with the tools' user interfaces, such as the readability of the written texts, the applicability of the presented icons and the overall user-friendliness of the evaluated tool. Further, because our participants are software engineers who have experience dealing with software development, analysis and testing tools and techniques, we give special attention to evaluating the features that are related to agile management methodologies and the ability to track progress at project and iteration levels. The results of our study can help the designers of these tools to identify the usability drawbacks of their systems and adjust them to fulfill the demands of the target users by highlighting the strengths and weaknesses of each tool from a usability engineering perspective. To the best of our knowledge, we have conducted the first research study that focuses on addressing the usability limitations of agile software project management tools based on a systematic approach that takes the subjective views of HCI experts and software engineers into consideration while utilizing the capabilities of a widely known usability assessment tool.

## 2 Literature Review

While taking into account the specific characteristics of software systems and the expectations of their users, many usability evaluation frameworks have been proposed to address the usability shortcomings of software applications in domains including banking, educational and gaming contexts [1–6]. While there are many usability parameters that could be considered for developing comprehensive usability evaluation criteria and frameworks, which might be related to either the properties of user interfaces or the cognitive abilities of target users, we believe that these parameters should not be considered equally important for all application types; some of them could be

valued over others depending on the domain requirements of target systems. In [1], for instance, the factors that increase learners' motivation to learn, the properties of instructional material and the subjective opinions of academics were utilized to develop a usability evaluation method for e-learning software systems. Similarly, researchers in [2] based the categorization of usability evaluation metrics specifically developed for massively multi-player online role-playing games on game-specific properties.

Prior studies have also highlighted the importance of reducing the development time of software systems, increasing the level of collaboration between members of development teams and achieving optimal allocation of product resources [7–10].

For globally distributed and co-located teams, the results presented in [8, 11] have shown that most of these teams use traditional communication, monitoring and task tracking tools (e.g., wikis, e-mails and instant messaging applications) instead of utilizing the software project management tools that were specifically designed to facilitate the coordination of distributed development work. Azizyan et al. found that the ease of use of agile project management tools is the most important factor impacting the adoption of these tools [12]. Silva et al. recommended studying the factors that would strengthen these tools and thus encourage software developers to take advantage of them [11]. In agile software project management contexts, after evaluating the ease of use of a number of software project management tools in [13], researchers have correlated the number of features provided by the evaluated tools with their ease of use and found that the tools with fewer features received higher ratings. However, the literature lacks comprehensive and systematic empirical evaluations of the usability of agile software project management solutions. There is also a lack of comprehensive usability evaluation criteria that could be taken as a basis for assessing the usability of these solutions from a project management perspective.

### 3 Research Methodology

We utilized a usability testing software called Morea to conduct a task-based usability analysis. We defined an evaluation checklist that was specifically developed for measuring the usability of software project management tools. We also used a pre-session questionnaire designed to help us understand our participants' experience in the software industry and their overall familiarity with our chosen tools. We also took advantage of the System Usability Scale (SUS) questionnaire to understand the overall usability of each tool included in our study. Each participant was also asked to answer a 5-point Likert scale questionnaire that was specifically developed to measure the usability of each tool from a software project management perspective. The statements included in this questionnaire are: to what extent do you agree that the use of the tool would help your development team to achieve its goals? (**Q1**); to what extent do you agree that the use of the tool would increase the level of collaboration between your team members? (**Q2**); to what extent do you agree that the usability problems that you observed would not have negative effects on the overall workflow within your team? (**Q3**); to what extent do you agree that the tool is easy to use? (**Q4**); to what extent do you agree that the tool would help in improving the productivity levels of your team members? (**Q5**); to what extent do you think that the tool would

help distributed teams to coordinate their work? (Q6); and to what extent do you agree that the tool would help in managing complex and large software systems? (Q7). Figure 1 summarizes our research methodology.

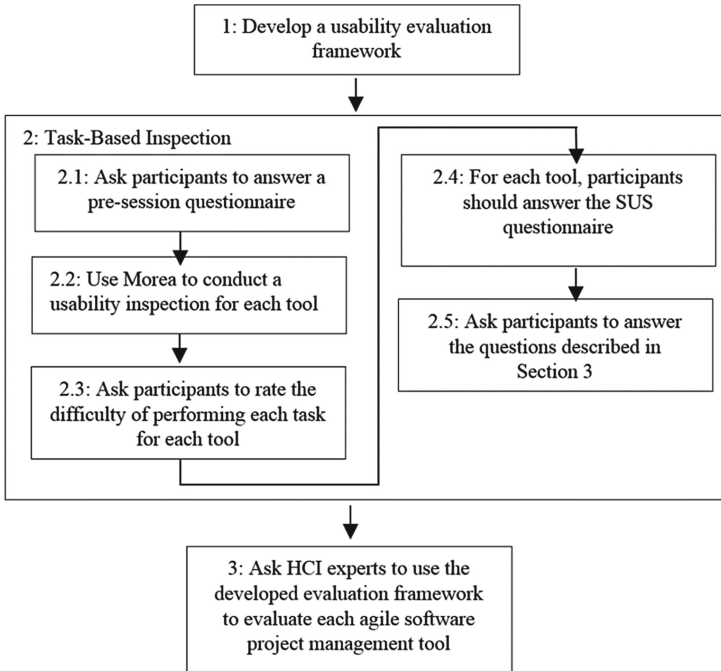


Fig. 1. Our research methodology

## 4 Usability Evaluation Criteria

The interface of each of the four selected agile project management tools was examined using Nielsen’s heuristics. From a project management perspective, some other heuristics were defined and have been mapped to the different project management-specific tasks. Each agile project management tool was examined by selecting its key tasks and examining its interface using the specified heuristics. This framework has grouped 41 defined metrics into different categories. These metrics were used by the five HCI experts to evaluate the different tools and their interfaces as shown in Table 5 which illustrates the overall heuristic evaluation of the four selected tools.

1. *Immediate Feedback*: The ability to provide appropriate feedback in the different cases while performing different tasks within a reasonable time. Two metrics were defined under this category: displaying error messages at the right time (A1) and presenting appropriate feedback based on explicit user actions (A2).
2. *Real-world mapping*: The ability of the tool to map and reflect real-world project management and development workflow. For this category, five metrics were

defined: the elimination of information irrelevant to software project management (**B1**), the usage of software project management metaphors (**B2**), the usage of user-oriented terms in the interface (**B3**), the usage of agile terminologies (**B4**) and the appropriate reflection of real-world software development workflow (**B5**).

3. *Consistency and standardization*: The ability to use standard terminologies and organization to avoid confusing the users with different terms and actions. Four metrics were defined under this category: the use of consistent in-tool and across-tools naming (**C1**), the use of consistent layout of interface elements (**C2**), the use of consistent colors across the user interface (**C3**) and the consistent alignment between user interface elements and documentation/help (**C4**).
4. *Ease of use and learnability*: The degree to which a tool can be used by the intended users to effectively achieve the required task with fewer errors/failures. Eight metrics are grouped under this category: the ease of mastering the software project management tool (**D1**), the ease of learning by different classes of users (e.g., project managers vs. team members) (**D2**), the ease of tracking the progress of other members (**D3**), the ease of task distribution (**D4**), the ease of changing the status of created tasks (e.g., from “in progress” to “completed”) (**D5**), the ease of locating burndown charts (**D6**), the ease of involving customers or clients in the development process (**D7**), the ease of specifying the complexity of each work item (**D8**) and the ease of navigating between different software project artefacts (**D9**).
5. *Layout and organization*: The ability to use a reasonable layout and logical grouping of the different project artefacts, which can help different users to navigate easily between different options and pages. Six metrics are defined under this category: the existence of reasonable grouping of project related artefacts (**E1**), reasonable grouping of sprint related artefacts (**E2**), reasonable grouping of team related artefacts (**E3**), logical grouping of menu options (**E4**), logical ordering of menu options (**E5**) and logical depth of menu options (**E6**).
6. *Flexibility*: Providing different options that allow users to handle and preform a given task easily. We consider whether the tool allows for a flexible arrangement of teams (**F1**), flexible reassignments of roles within and across projects (**F2**), flexible adjustment of the status of tasks (e.g., in progress or completed) (**F3**), flexible prioritization of software requirements according to their importance (**F4**) and flexible management of time (**F5**).
7. *Streamlining the experience and visibility*: The ability of a tool to match real-world scenarios and communicate the context of the situation. We consider the support for user-customized profiles (**G1**), the appropriate support for recognition rather than recall (**G2**), the efficiency of task completion (**G3**), the availability of visually appealing user interface designs (**G4**), the visibility of visual interface elements (**G5**) and the visibility of the roles of different team members (**G6**).
8. *Clarity*: Enabling users to interact with the tool and distinguish its tasks easily without causing confusion. Under this category, we examine whether there is a clear distinction between the tasks and user stories throughout the user interface (**H1**), a clear distinction between bugs, epics and defects (**H2**), a clear distinction between the roles of software-development team members (**H3**) and clear navigation options throughout the software project management tool (**H4**).

## 5 Experimental Evaluation: Task-Oriented Usability Inspection

Eight participants were recruited to perform 17 tasks on the four chosen agile software project management tools (JIRA, VersionOne, AgileZen and ZebraPlan). Although agile software project management tools differ in the services they offer and have many features, we chose tasks that we believe most software engineers who work in development teams would utilize. Verifying admin home pages (**T1**), creating a project (**T2**), creating a member (**T3**), creating a user story (**T4**) and creating a task (**T5**) were the first five tasks that we asked our participants to perform. Our participants also tried creating sprints (**T6**), creating releases or versions (**T7**), assigning team members to tasks (**T8**), customizing the settings of their profile pages (**T9**) and viewing burndown charts (**T10**). They were also asked to specify the length of sprints in days or weeks (**T11**), access project conversation rooms (**T12**), mark tasks as completed (**T13**), change the priority of a task (**T14**), track the progress of team members (**T15**), place a comment on someone's work (**T16**) and log in as clients and give feedback (**T17**). During each session, two observers were taking notes and tracking the progress of the participants. To identify usability problems and report users' concerns while they were experimenting with the tools, the think-aloud protocol was utilized. Further, we intentionally decided not to ask participants to follow a particular order while performing the required tasks to avoid the effect resulting from using a difficult tool on an easier one. Participants were also asked to rate the difficulty of each task using a 5-point Likert scale.

## 6 Results

**Pre-session Questionnaire.** At the start of each session, the participants were asked to complete a short pre-session questionnaire so we could assess their familiarity with the software engineering domain and software project management tools. The questionnaire was composed of six closed-format questions. Most of the participants have worked as software engineering practitioners in the software industry. Six of our participants had worked in the industry for a couple of months and two of them had worked for one to three years. All the participants who had worked in software engineering domains were responsible for programming and writing software codes, while only one participant was responsible for testing. In addition, four participants worked as requirement analysts, designers and project managers. Most of the participants have used software project management tools to coordinate their development work; however, most of them indicated that they used Microsoft Project and web-based tools (e.g., Google spreadsheets and wikis) to coordinate their work. Of the eight participants, two had used VersionOne whereas only one participant indicated that she had used JIRA, AgileZen and ZebraPlan.

**Task Completion.** For all 17 of the above-mentioned tasks, Table 1 illustrates the task success rates among the participants for the four agile software project management tools. For JIRA, for instance, all of the participants successfully completed four tasks out of seventeen without facing any difficulty. All participants were able to figure out how to

**Table 1.** Task completion rates among our eight participants

	JIRA	VersionOne	AgileZen	ZebraPlan
<b>T1</b>	8/8	7/8	8/8	8/8
<b>T2</b>	8/8	5/8	8/8	8/8
<b>T3</b>	7/8	8/8	8/8	8/8
<b>T4</b>	4/8	0/8	8/8	7/8
<b>T5</b>	7/8	6/8	6/8	8/8
<b>T6</b>	7/8	6/8	7/8	7/8
<b>T7</b>	5/8	5/8	1/8	0/8
<b>T8</b>	7/8	8/8	5/8	7/8
<b>T9</b>	8/8	8/8	8/8	8/8
<b>T10</b>	7/8	6/8	7/8	8/8
<b>T11</b>	6/8	8/8	1/8	7/8
<b>T12</b>	1/8	8/8	2/8	0/8
<b>T13</b>	6/8	5/8	8/8	8/8
<b>T14</b>	7/8	7/8	7/8	8/8
<b>T15</b>	5/8	7/8	6/8	7/8
<b>T16</b>	8/8	6/8	7/8	3/8
<b>T17</b>	1/8	4/8	3/8	1/8

**Table 2.** The average numbers of mouse clicks and time our participants required to complete the tasks

Task	JIRA		VersionOne		AgileZen		ZebraPlan	
	TT	NMC	TT	NMC	TT	NMC	TT	NMC
<b>T1</b>	13.2	1.43	13.6	1.25	8	1	22.8	3.88
<b>T2</b>	34.5	5.71	112	22.1	10.7	1.38	45.1	10.6
<b>T3</b>	62.4	8.86	40.5	7.25	31.5	6	53.7	7.5
<b>T4</b>	116	20	124	22.5	54	10.1	58.9	8.38
<b>T5</b>	47.3	11.6	98.2	18.1	106	19.6	16.2	3.38
<b>T6</b>	69.8	11.6	82.9	11.4	88.4	18	45.3	11.9
<b>T7</b>	85.2	15.6	89.2	14.9	93.8	23.5	69.3	13.9
<b>T8</b>	28.2	6.13	52.9	10.3	46.2	12.8	47.5	13.7
<b>T9</b>	32.3	6.5	35.9	5.63	21.2	3.25	18.7	3.75
<b>T10</b>	58.1	12.9	54.5	10.8	65.5	19.6	15.4	3.75
<b>T11</b>	122	27.3	39.4	6.88	43.2	9.38	31.9	9.88
<b>T12</b>	84	16.3	25.4	4.13	55.7	11.4	51.1	10.4
<b>T13</b>	89.5	23.3	143	28.6	52.8	12.4	60.1	13.6
<b>T14</b>	51.3	10.8	59	11.9	38.4	8.38	21.6	6
<b>T15</b>	71.6	14.5	81.5	12.4	41.7	8	54.4	9.63
<b>T16</b>	15.5	3.38	44.	7.57	30	7.75	39.8	9.13
<b>T17</b>	55.4	9.38	72.1	8.25	48.8	9.63	34.2	6.38
Mean	61	12	68	11.9	49.2	10.7	40.4	8.58

complete the third, eighth, ninth, eleventh and twelfth tasks using VersionOne. AgileZen showed a better success distribution in comparison with JIRA and VersionOne in that six out of the 17 tasks were completed successfully by all eight participants. ZebraPlan obtained the greatest success distribution among the four project management tools as all participants were able to successfully complete 10 out of the 17 tasks.

**Time on Task (TT).** For the four software project management tools evaluated in this paper, Table 2 shows the average time spent by our participants to complete each task (in seconds).

**Number of Mouse Clicks (NMC).** We use this metric to help us determine how easily a user can accomplish a basic task the first time they use the tool. We measured navigability by calculating the number of mouse clicks required to complete each task using each of the four evaluated tools (see Table 2). Depending on the difficulty of the task and how easily a user could navigate through the user interface of each tool, we observed variations in the average numbers of mouse clicks our users needed to complete each task using the four tools.

**System Usability Scale (SUS).** This questionnaire is composed of ten statements that are scored on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). In SUS, the statements cover a variety of aspects of system usability such as learnability, reliability and validity, thus providing us with high-level measurement of the usability of each tool. According to [14], the average SUS score is 68 %. Table 3 demonstrates the average SUS scores for the four tools. AgileZen and ZebraPlan achieved very high average SUS scores of 70.63 % and 80.31 %, respectively. These results suggest positive perceptions of the usability of the two systems. On the other hands, the average SUS scores for JIRA and VersionOne were similar (44.69 % for JIRA and 42.19 % for VersionOne). The SUS scores for both JIRA and VersionOne were below the average acceptable SUS score.

**Second Post-session Questionnaire.** We included questions to indicate whether the use of each of the four evaluated tools would help streamline development and management activities in agile teams (see Sect. 3). Using 5-point Likert scale questions, we asked our participants to rate the extent to which the utilization of each of the four software project management tools would help fulfill the objectives of development teams, facilitating discussions and interactions between team members and managing large and complex software systems. Our questions also addressed the applicability of using each tool to coordinate distributed development projects and reflect the overall workflow of development teams in real-world scenarios. After mapping each answer to a score (e.g., *strongly agree* answers worth 5 points, *strongly disagree* answers worth 1 points, etc.) and calculating the means of these scores, we were able to measure the degrees of satisfaction of our participants for each agile management tool (see Table 4).

For ZebraPlan, all of our participants agreed that it is easy to use and expected it to help development teams achieve their goals. On the contrary, our results show that JIRA and VersionOne were more complex to use compared with ZebraPlan and AgileZen. When asked about the positive effects the tools would have on the productivity levels of team members, our results show that there were slight differences between the calculated



**Table 3.** Results of the SUS questionnaire

	JIRA	VersionOne	AgileZen	ZebraPlan
Minimum	25.00	17.50	22.50	50.00
Maximum	70.00	70.00	97.50	97.50
Mean	44.69	42.19	70.63	80.31
Standard Dev.	16.12	17.50	23.37	16.17

**Table 4.** Results of our second post-session questionnaire

	JIRA	VersionOne	AgileZen	ZebraPlan
<b>Q1</b>	3.25	3.25	3.75	4.5
<b>Q2</b>	3.75	3.88	2.88	3.63
<b>Q3</b>	2.75	2.63	4.13	3.63
<b>Q4</b>	2.75	2.63	3.88	4
<b>Q5</b>	3.38	3.63	3.75	4
<b>Q6</b>	3.25	3.25	3	3.63
<b>Q7</b>	2.88	3.38	2.63	3.5

means for all the tools although ZebraPlan and AgileZen took the first and second positions, respectively. By linking this finding with the level of complexity reported for each tool, we note that the tools that were observed as easiest to use were expected to have the greatest positive effects on the productivity levels of software engineers.

From the results presented in Table 4, we can also observe that JIRA and VersionOne obtained the lowest average scores for the third question, indicating that most participants expected the usability problems they faced to have negative effects on the overall workflow of development work in agile teams. For managing large and complex systems, most of the participants preferred using VersionOne or ZebraPlan. Our findings also show that participants found VersionOne to be helpful in managing communications and discussions between team members in development teams. Considering that ZebraPlan obtained the highest mean scores for five of the seven questions and was reported the easiest to use, our results suggest that the usability of this software project management tool would have significant impact on the overall progress and coordination of development work in teams that follow agile methodologies.

**Observed Usability Problems.** Generally, due to the variety of artefacts used among developers who work in agile teams, we note that the four tools differ in the ways they present and organize the functions that each user can take advantage of. For instance, when adding a task, not all of the tools ask users to specify the user story to which the task belongs. Further, when specifying a type of task, some of the tools present the user with a list of options to choose from whereas others ask the user to type in the details of each backlog item that he/she would like to add. To prioritize user stories or development tasks, some of the tools ask the user to choose whether the priority of a task is low, medium or high, while others ask users to either type in the priorities that they want to specify or employ drag-and-drop features. Thus, by considering the fact that numbers can also be used for specifying task priorities, we note that typing the values of these

priorities could lead to team members inconsistently specifying these values, which might slow the overall progress of development works. To change the status of task, we observed that some tools allow users to indicate whether a task has started, is in progress or has been completed, whereas others present checkboxes to allow users to indicate the percentage of completion of a given task as either 0 % or 100 %. We also observed that due to inconsistencies in the presentation and structure of functions inside some of the evaluated tools, some participants struggled to find the appropriate button and/or link to perform some of the tasks. For example, although each project has a set of sprints with some user stories that might be further divided into tasks, some of the evaluated tools do not clearly organize these pieces of information based on this hierarchy or do not present users with clear indications to allow them to find the required information easily. After spending a while trying to add a backlog item, one of the participants said, “The navigation menus are not helpful, I remember I saw the button I need but I have no idea where it is” (P4). Other participants also indicated that they had to remember so much information or practice and learn how to use some of the functions many times in order to efficiently perform the required tasks. Therefore, we recommend attaching all the artefacts related to a specific project consistently throughout agile software project management tools. We note that participants faced problems related to the visibility and placement of interface elements (e.g., some tools use grey to color clickable URLs). Furthermore, some participants failed to complete certain tasks because some hyperlinks

**Table 5.** Ratings reported by our HCI experts based on our defined criteria (see Sect. 4)

	S1	S2	S3	S4		S1	S2	S3	S4
<b>A1</b>	5	5	5	4	<b>E1</b>	3	4	4	2
<b>A2</b>	5	1	1	1	<b>E2</b>	4	4	1	3
<b>B1</b>	5	5	5	5	<b>E3</b>	3	3	4	2
<b>B2</b>	5	5	5	4	<b>E4</b>	4	2	4	3
<b>B3</b>	5	5	5	5	<b>E5</b>	4	4	4	3
<b>B4</b>	5	5	5	4	<b>E6</b>	4	4	4	4
<b>B5</b>	4	4	5	4	<b>F1</b>	5	5	3	4
<b>C1</b>	4	3	4	3	<b>F2</b>	4	5	3	5
<b>C2</b>	4	3	3	3	<b>F3</b>	5	5	3	5
<b>C3</b>	5	5	5	5	<b>F4</b>	5	5	2	5
<b>C4</b>	5	5	5	5	<b>F5</b>	4	5	3	5
<b>D1</b>	2	3	5	4	<b>G1</b>	5	5	4	3
<b>D2</b>	2	3	5	4	<b>G2</b>	4	3	4	5
<b>D3</b>	4	4	5	5	<b>G3</b>	3	3	4	5
<b>D4</b>	5	4	4	5	<b>G4</b>	5	5	3	5
<b>D5</b>	5	5	5	5	<b>G5</b>	3	3	2	5
<b>D6</b>	5	5	4	2	<b>G6</b>	3	4	4	5
<b>D7</b>	5	5	2	5	<b>H1</b>	4	4	5	4
<b>D8</b>	1	1	1	1	<b>H2</b>	5	3	1	4
<b>D9</b>	2	2	5	2	<b>H3</b>	3	3	2	5
-	-	-	-	-	<b>H4</b>	3	2	4	5

did not appear as clickable. In some of the evaluated tools, our participants also faced difficulty in tracking the progress of team members or previously created tasks, either because they could not find the task boards or product backlogs in the corresponding tools or forgot where to find them. It is also worth mentioning that some participants queried search engines to figure out where to find burndown charts, add members to their teams and specify the start and end dates of sprints.

**Heuristics Evaluation.** Table 5 presents the results of the usability ratings reported by five HCI experts. The evaluation is based on the criteria define in Sect. 4. In this table S1, S2, S3 and S4 represent JIRA, VersionOne, AgileZen and ZebraPlan, respectively.

## 7 Conclusion

In agile project management contexts, the involvement of human, time, financial and organization-specific factors increases the burden on usability practitioners to study the factors that could improve the workflow of agile teams in software project management tools without increasing the complexity of these tools. In this research effort, we utilize a number of qualitative and quantitative usability engineering methods to identify the major and minor drawbacks of four widely used agile software project management tools. We believe that the experimental findings and the usability evaluation framework presented in this paper can help software development companies to select their tools. We also expect our results to help designers of agile project management tools to identify the shortcomings of their solutions and improve them to suit the requirements of co-located and distributed teams that follow agile software development methodologies.

## References

1. Zaharias, P.: A usability evaluation method for e-learning: focus on motivation to learn. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 1571–1576. ACM (2006)
2. Song, S., Lee, J.-H., Hwang, I.: A new framework of usability evaluation for massively multi-player online game: case study of “world of warcraft” game. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4553, pp. 341–350. Springer, Heidelberg (2007)
3. Lettner, F., Holzmann, C.: Usability evaluation framework. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2011, Part II. LNCS, vol. 6928, pp. 560–567. Springer, Heidelberg (2012)
4. Chen, S.Y., Macredie, R.D.: The assessment of usability of electronic shopping: A heuristic evaluation. *Int. J. Inf. Manage.* **25**(6), 516–532 (2005)
5. Dybkjær, L., Børnsen, N.O.: Usability evaluation in spoken language dialogue systems. In: Proceedings of the workshop on Evaluation for Language and Dialogue Systems, vol. 9, p. 3, Association for Computational Linguistics (2001)
6. Alhumoud, S., Alabdulkarim, L., Almobarak, N., Al-Wabil, A.: Socio-cultural aspects in the design of multilingual banking interfaces in the arab region. In: Kurosu, M. (ed.) Human-Computer Interaction. LNCS, vol. 9171, pp. 269–280. Springer, Heidelberg (2015)

7. Sathi, A., Morton, T.E., Roth, S.F.: Callisto: An intelligent project management system. *AI Mag.* 7(5), 34 (1986)
8. Katsma, C., Amrit, C., van Hillegersberg, J., Sikkel, K.: Can agile software tools bring the benefits of a task board to globally distributed teams? In: Oshri, I., Kotlarsky, J., Willcocks, L.P. (eds.) *Global Sourcing 2013. LNBP*, vol. 163, pp. 163–179. Springer, Heidelberg (2013)
9. Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>. Accessed 2 March 2016
10. Al-Ani, B., Wang, Y., Marczak, S., Trainer, E., Redmiles, D.: Distributed developers and the non-use of web 2.0 technologies: a proclivity model. In: 2012 IEEE Seventh International Conference on Global Software Engineering (ICGSE), pp. 104–113. IEEE (2012)
11. da Silva, F.Q.B., Costa, C., Franca, A.C.C., Prikladinicki, R.: Challenges and solutions in distributed software development project management: a systematic literature review. In: 2010 5th IEEE International Conference on Global Software Engineering (ICGSE), pp. 87–96. IEEE (2010)
12. Azizyan, G., Magarian, M.K., Kajko-Matsson, M.: Survey of agile tool usage and needs. In: Agile Conference (AGILE), 2011, pp. 29–38. IEEE (2011)
13. Azizyan, G., Magarian, M., Kajko-Mattson, M.: The dilemma of tool selection for agile project management. In: 7th International Conference on Software Engineering Advances (ICSEA 2012), pp. 605–614 (2012)
14. Sauro, J.: Measuring usability with the system usability scale (sus). <http://www.measuringusability.com/sus.php>. Accessed 5 March 2016