Effectiveness Comparison Between Kanban and Scrum on Software Development Projects

Farnaz Ganjeizadeh*, Helen Zong, Pinar Ozcan, and Erik Olivar

Department Of Engineering
California State University, East Bay
Hayward, CA, 94542, USA

ABSTRACT

The challenges in managing software development in technology oriented companies have led to different project management methodologies such as Waterfall and Agile approach. Agile methodologies are a group of software development methods that are incremental and iterative. These methods are more effective and important for project management. For the companies that develop complex software, using agile methodologies is particularly important. Kanban and Scrum are two powerful agile project management approaches in software development. The objective of Scrum and Kanban is to help companies work more effectively by telling what to do, how to manage time, how to set-up the team, and finally how to optimize the process. Although these methodologies are significant to companies, a review of literature indicates that there is lack of statistical evidence about which methodology is more effective in terms of budget handling, risk control, quality of the project, available resources, clear project scope and schedule handling. This research investigated the details of Scrum and Kanban and the comparative analysis between them in terms of budget, schedule, risk, resources, scope and quality.

1. INTRODUCTION

Kanban and Scrum are two powerful and the most common agile project management approaches in software development. Although there has been a debate for years about these project methodologies, there is no statistical supporting analysis in the existing literature to demonstrate which methodology is more efficient. Therefore, the focus of this research is to provide a statistical evidence to determine if there is a difference between these two methodologies in terms of budget handling, risk control, quality of the project, available resources, clear project scope, and schedule handling in software development projects.

The *Project Management Book of Knowledge* (*PMBOK 4.0*) offers a 6-pointed star model to represent the success factors in a project and show the project inputs and outputs and their separation. These factors are scope, budget, and schedule as inputs; and risk, quality, and resources as outputs.

In this study, the 6-pointed star model is used to demonstrate the effectiveness of Kanban and Scrum. Firstly, the data is collected via a web-based survey. The survey questions are designed to reflect the input/output factors listed above. Secondly, the correlation between these factors and success of a project is verified by two-tailed Pearson correlation analysis. Finally, Kanban and Scrum methodologies are tested by independent two-sample t-test in terms of budget, scope, schedule, risk, quality and resources to statistically verify that one methodology is superior to the other.

2. LITERATURE REVIEW

2.1. OVERVIEW OF SCRUM THEORY

Sutherland and Schwaber [1] describe Scrum as a lean approach to software development. It is based on empirical process control theory and is an iterative and incremental project management methodology used to control risk and optimize the predictability of a project. Three important factors in the Scrum process are transparency, inspection and adaptation [2]. Transparency requires that the process be visible to everyone who is involved in the project; inspection requires that Scrum users inspect Scrum artifacts frequently to detect problems in early stages; and adaptation requires

* Corresponding author: Tel.: (510) 885-4862; Fax: (510) 885-2678; E-mail: farnaz.ganjeizadeh@csueastbay.edu

that the process be adjusted to avoid further problems if an inspector determines that some aspects of the project are unacceptable and outside the project scope. These three factors should be applied at different phases of deployment of the project. Scrum has been applied in various software development projects as described in [3] and [4] and has been used in conjunction with integration models as described in [5].

2.2. Overview of Kanban Theory

Kanban is another project management methodology for software development that emphasizes on just-in-time delivery while avoiding software developers. The main focus of Kanban is to state when the work will be implemented accurately by prioritizing tasks and defining workflow and lead-time to delivery [6]. It also shows the most valuable tasks explicitly that need more attention and energy to reduce the risk of uncompleted tasks and to increase the flexibility among the project. For instance, project developers do not implement unnecessary features, do not write more specifications than they can code, do not write more code than they can test, and do not test more code than they can deploy [7]. Additional discussions of Kanban applications in software development are found in [8] and [9], and a discussion of how Kanban affects team performance is discussed in [10].

2.3. Similarities between Kanban and Scrum

Kniberg and Skarin [11] stated that Kanban and Scrum are similar in the following areas: being lean and agile, breaking the work into small pieces, having self-organized teams, focusing on the delivery of releasable software early and often, adapting changes quickly, limiting work-in-process (WIP), using pull scheduling, and using transparency.

In addition, Keogh [12] explains that both methods contain feedback and improvement mechanisms and have a clear focus on scope. They also put the importance of value on the project and achieve the valuable deliveries. Moreover, Ladas [13] discusses how aspects of Kanban and Scrum can both be applied to a lean software development project. Although Kniberg, Skarin, and Keogh explain the similarities of Kanban and Scrum in detail, they do not exemplify any similarity in budget, schedule and risk handling.

2.4. Differences between Kanban and Scrum

Kniberg and Skarin [11] summarize all the main differences between Kanban and Scrum. Table 1 shows these differences between the two methodologies. Although Kniberg explains the differences between Kanban and Scrum in detail, the author does not emphasize their effects on the success of projects in terms of input/output factors. Among the differences that are listed in Table 1, the major ones are time, capacity, and prioritization.

Table 1. Differences between Kanban and Scrum.

Scrum	Kanban
Timed-boxed iterations prescribed.	Time-boxed iterations optional. Can have separate cadences for planning, release, and process improvement. Can be event-driven instead of time-boxed.
Team commits to a specific amount of work for this iteration.	Commitment optional.
Uses velocity as default metric for planning and process improvement.	Uses lead time as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. Specialist teams allowed.
Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burn down chart prescribed.	No particular type of diagram is prescribed.
WIP limited indirectly (per sprint).	WIP limited directly (per workflow state).
Estimation prescribed.	Estimation optional.
Cannot add items to ongoing iteration.	Can add new items whenever capacity is available.
A sprint backlog is owned by one specific team.	A Kanban board shared by multiple teams or individuals.
A Scrum board is reset between each sprint.	A Kanban board is persistent.
Prescribes a prioritized product backlog.	Prioritization is optional.

3. RESEARCH METHODOLOGY

3.1. DATA COLLECTION

The research was conducted via a web-based survey. This survey was open for a one-month period from April 2012 to May 2012 to employees who are involved in software development projects and sent via email and Linkedin.com to companies using Scrum or Kanban as a project management methodology. The questions were designed to analyze the effectiveness of these methodologies on the software projects in terms of risk, quality, resources, budget, scope and schedule. Table 2 shows the factors and related questions on the survey used to calculate which methodology is more effective than the other.

Table 2. Factors and related survey questions.

Factor	Question
Schedule	Project teams are aware of the project status
	Project is delivered on time according to schedule
	Project teams can adapt the changes quickly
Scope	Project usually has a well-defined scope
	Methodology is effective to make the scope clearer
Budget	Project is delivered within budget
	Project provides a good return on investment (ROI)
Risk	Project risks and opportunities are managed
	Business objectives are met
Resources	Human and material resources are mostly available
	Teams can work well together to achieve expected results
Quality	Quality requirements are met
	Client satisfaction is met
	Project is successful overall

3.2. DEMOGRAPHICS OF SURVEY RESPONDENTS

A total of 35 responses were collected for the survey, and 24 of the respondents specified their sectors. Respondents are mainly working in IT and consultancy sectors. Table 3 displays the categorization of the respondents' companies.

Table 3. Sectors respondents' companies are in.

Sector	Number of Respondents
Information technology	6
Consultancy	4
Education	3
E-commerce	2
Bank and finance	2
Warranty	2
Web services	1
Tourism	1

All of the 35 respondents shown above specified the company size as shown in Table 4. Survey shows that the majority of respondents were within the range of 50-100 and 100-500. As the data indicates, 37.1% of the respondents are working in companies whose size is 50 to 100. It is followed by 100 to 500 with 28.6% response percent.

Table 4. Size of the company.

Size of Company	Percentage of Respondents
Less than 50	25.7%
50-100	37.1%
100-500	28.6%
More than 500	8.6%

Eight of the 35 respondents are project managers, seven software engineers, and three assistant project managers as shown in Table 5. Additionally, web developers and IT managers also took this survey as shown in the table below.

Table 5. Position of each respondent.

Position	Number of Respondents
Project manager	8
Software developer	7
Assistant project manager	3
Engineer	3
Web developer	3
IT manager	3
Co-founder	2
Analyst	2
Product manager	1
System administrator	1
Intern	1

In addition, respondents answered how many years of work experience they have, with the data listed in Table 6. It should be noted that one respondent did not answer this question. From the results, 32.4% of respondents stated they worked less than 2 years, 32.4% stated they worked from 5-10 years, and 26.5% stated they worked 2-5 years.

Table 6.Years of work experience.

Years of work experience	Percentage of Respondents
Less than 2	32.4%
2-5	26.5%
5-10	32.4%
More than 10	8.8%

Table 7. Number of people working on software project.

Number of people	Percentage of Respondents
Less than 10	42.9%
10-20	45.7%
21-50	5.7%
More than 50	5.7%

Respondents were also asked regarding the number of people working in software projects as shown in Table7. It is noted that 45.7% of the respondents stated that there are 10 to 20 personnel working in software projects, where 42.9% stated there were less than 10 personnel.

The emphasis of the application of our methodology was conducting comparative analyses between Scrum and Kanban. The analysis of the survey data shows that 60% of the respondents (21 respondents) are using Scrum as a project management methodology in their software development projects. 40% of them (14 respondents) are using Kanban as listed in Table 8.

Table 8. Number of respondents using Scrum and Kanban.

Methodology	Number of Respondents
Scrum	21
Kanban	14

3.3. SURVEY RESULTS

The data set was divided into two main subsets: data related to Kanban projects and data related to Scrum projects. For each data subset, the number of answers to the related questions for schedule, budget, scope, risk, resources and quality was multiplied by the weights assigned to the Likert scale as shown in Table 9.

Table 9. Likert Scale.

Response	Weight
Strongly Disagree	1
Disagree	2
Neutral	3
Agree	4
Strongly Agree	5

All 35 respondents answered the questions listed in Table 2. The percentage and number of respondents for each answer along with the average rating are provided in Table 10.

Table 10. Survey responses.

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Rating Average	Response Count
Project usually has well defined scope	0.0% (0)	31.4% (11)	20.0% (7)	34.3% (12)	14.3% (5)	3.31	35
PM methodology effective to make scope clearer	2.9% (1)	2.9% (1)	20.0% (7)	45.7% (16)	28.6% (10)	3.94	35
Project teams are aware of project status	2.9% (1)	5.7% (2)	20.0% (7)	31.4% (11)	40.0% (14)	4.00	35
Project teams can adapt changes quickly	2.9% (1)	11.4% (4)	8.6% (3)	48.6% (17)	28.6% (10)	3.89	35
Project is delivered on time according to schedule	5.7% (2)	11.4% (4)	25.7% (9)	34.3% (12)	22.9% (8)	3.57	35
Project is delivered within budget	5.7% (2)	5.7% (2)	22.9% (8)	45.7% (16)	20.0% (7)	3.69	35
Human/material resources are mostly available	2.9% (1)	14.3% (5)	31.4% (11)	40.0% (14)	11.4% (4)	3.43	35
Project risks and opportunities are managed	5.7% (2)	2.9% (1)	34.3% (12)	40.0% (14)	17.1% (6)	3.60	35
Teams work well together to achieve expected results	2.9% (1)	0.0% (0)	5.7% (1)	40.0% (14)	51.4% (18)	4.37	35
Quality requirements are met	2.9% (1)	2.9% (1)	17.1% (6)	48.6% (17)	28.6% (10)	3.97	35
Client satisfaction is met	2.9% (1)	8.6% (3)	20.0% (7)	45.7% (16)	22.9% (8)	3.77	35
Business objectives are met	2.9% (1)	0.0% (0)	11.4% (4)	51.4% (18)	34.3% (12)	4.14	35
Project provides good return on investment	2.9% (1)	5.7% (2)	17.1% (6)	40.0% (14)	34.3% (12)	3.97	35
Project is successful overall	2.9% (1)	0.0% (0)	2.9% (1)	57.1% (20)	37.1% (13)	4.26	35

4. COMPARATIVE ANALYSIS

4.1. Comparison between Mean Scores Of Kanban And Scrum Projects

After identifying which respondents used which methodology and separating them into a Kanban data set and a Scrum data set, the mean scores for risk, resources, budget, schedule, scope and quality were calculated to proceed with t-testing. The mean scores were calculated by multiplying the Likert scale weight by the total number of answers to that question. The questions in the same factor category such as schedule, budget, etc., were added to each other to find the final mean score. Table 11 shows the comparison between mean scores for those factors in Scrum and Kanban projects. It is apparent from a comparative analysis of the means that the application of the two methods reveals that there is no significant difference between the means at our selected domain.

Table 11. Comparison between Kanban and Scrum mean scores.

Factor	Kanban Mean Scores	Scrum Mean Scores
Schedule	12.14285714	10.85714286
Scope	7.357142857	7.333333333
Budget	7.571428571	7.857142857
Risk	7.642857143	8.047619048
Resources	7.857142857	7.761904762
Quality	12.07142857	11.95238095

4.2. Correlation Between Success Of The Project And Input/output Factors

To empirically prove that factors listed in the *Project Management Body of Knowledge (PMBOK 4.0)* are correlated to the quality and success of the software project, two-tailed Pearson Correlation was applied. These input/output factors are budget, schedule, risk, resources, and scope. Table 12 shows the Pearson's r-values and their association with these factors in software development projects that are using Kanban or Scrum. It should be noted that the correlation of the factors above are dependent on the level of significance. For example, with schedule handling, Scrum results indicate a strong positive correlation with r = 0.751, while Kanban results show a weak positive correlation with r = 0472. The correlation data for other factors are tabulated in the same manner in Table 12. The next step was to perform hypothesis testing in order to evaluate the difference between two approaches.

Table 12. Correlation between success of the project and input/output factors for both Scrum and Kanban.

	Scrum		Kanban	
Factor	Correlation/Level of Significance (N = 21)	Association	Correlation/Level of Significance (N = 14)	Association
Clear scope	r: 0.769275865 p: 0.0000458 **	Strong positive	r: 0.89161981 p: 0.00001847**	Strong positive
Schedule handling	r: 0.750558166 p: 0.00008874 **	Strong positive	r: 0.4726929 p: 0.08784311	Weak positive
Budget control	r: 0.76989824 p: 0.0000447 **	Strong positive	r: 0.334358346 p: 0.24264576	Weak positive
Risk handling	r: 0.825562088 p: 0.00000407 **	Strong positive	r: 0.563102275 p: 0.03602292*	Weak positive
Available resources	r: 0.7032335 p: 0.00037639 **	Weak positive	r: 0.343697938 p: 0.22891427	Weak positive

** Correlation is significant at 0.01 level

* Correlation is significant at 0.05 level

4.3. HYPOTHESIS TESTING

Hypothesis tests were then conducted to determine if there was a significant difference between the effectiveness of Kanban and Scrum on the factors of the project. The hypothesis tests aimed to validate which project management methodology is more efficient throughout the project for each factor. A hypothesis test for each factor was structured as illustrated in equation (1) and (2) listed below.

H_o: There is no difference between the mean (factor) scores for Kanban and Scrum in software development projects.

$$H_o: \beta = 0 \quad (1)$$

$$H_a: \beta \neq 0 \quad (2)$$

If p-level of two-sample t-test is greater than alpha = 0.05, then fail to reject H_o.
If p-level of two-sample t-test is less than alpha = 0.05, then reject H_o.

The sample sizes for Kanban and Scrum were 14 and 21, respectively. A summary of the statistical analysis is presented below in Table 13.

Table 13. Two-tailed distribution p-levels for each factor.

Factor being tested	**Two-tailed distribution p-level**
Schedule	0.14187
Scope	0.96333
Budget	0.61017
Risk	0.48588
Resources	0.86577
Quality	0.87356
Success	0.78059

From the table above, the p-levels for each of the factors being tested are greater than alpha (α) = 0.05. Thus, we fail to reject H_o under the 95% confidence level and infer than Scrum and Kanban have the same effectiveness level on the schedule, scope, budget, risk, resources, quality, and success of software projects.

5. CONCLUSION

There has been a debate for years about whether Scrum or Kanban is more efficient than the other. However, the literature review indicates that there is a lack of statistical evidence on this topic in the existing literature. This research was conducted to see if there is a significant difference between these two methodologies impacting the following six factors: budget handling, risk control, quality of the project, available resource, clear project scope, and schedule handling in software development projects. These factors were chosen as they were listed as six factors to be monitored and managed in the *Project Management Body of Knowledge (PMBOK 4.0)*.

In order to measure the effectiveness of Kanban and Scrum on software projects, a survey was designed to include various questions about the company, software projects, project management methodology, implementation, and feedback of the projects. The correlation analysis demonstrated that there is a positive association between these factors and success. Subsequently, independent two-sample t-tests were conducted to test the hypotheses. The results imply that there is no statistically significant difference between Kanban and Scrum in terms of risk, budget, schedule, resources, quality, scope and overall success of the projects.

Although this study exemplifies that one methodology is not significantly superior to the other one, companies should be aware of the differences in the practical implementation of these methodologies carefully. For future work, we should consider the impact of additional non-quantitative techniques, such as team commitment, work organization, schedule managing, allocation of resources, and visibility.

REFERENCES

[1] *Jeff Sutherland and Ken Schwaber: "The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process", Jeff Sutherland, pp. 11–19, 1993-2007.*

[2] *https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf (accessed September 2012).*

[3] *Chen Li, Marjan van den Akker, Sjaak Brinkkemper, and Guido Diepen: "An Integrated Approach for Requirement Selection and Schedulding in Software Release Planning", Requirements Engineering, Vol. 15, Issue 4, pp. 375–396, 2010.*

[4] *Marko Seikola: "The Scrum Product Backlog as a Tool for Steering the Product Development in a Large-Scale Organization", Aalto University School of Science and Technology, pp. 1–96, 2010.*

[5] *K. Lukasiewics and J. Miler: "Improving Agility and Discipline of Software Development with the Scrum and CMMI", IET Software, Vol. 6, Issue 5, pp. 416–422, 2012.*

[6] *D.J. Anderson: "What is the Kanban Method", Blue Hole Press, pp. 11–16, 2010.*

[7] *D.J. Anderson:"Theories for Agile Management in Agile Management for Software Engineering", Pearson Education, Inc., pp. 3–12, 2004.*

[8] *Muhammad Ovais Ahmad, Jouni Markula, and Markku Ovio: "Kanban in Software Development: A Systematic Literature Review, Software Engineering and Advanced Applications, pp. 9–16, 2013.*

[9] *Noureddine Kerzazi: "Kanbanize the Release Engineering Process", Release Engineering, pp. 9–12, 2013.*

[10] *Nilay Oza, Fabian Fagerholm, and Jurgen Munch: "How does Kanban Impact Communication and Collaboration in Software Engineering Teams?", Cooperative and Human Aspects of Software Engineering, pp. 125–128, 2013.*

[11] *http://www.infoq.com/minibooks/kanban-scrum-minibook (accessed June 14, 2012).*

[12] *http://lizkeogh.com/2011/11/20/scrum-and-kanban-both-the-same-only-different/ (accesseed June 14, 2012).*

[13] *Corey Ladas: "Scrumban: Essays on Kanban Systems for Lean Software Development", Modus Cooperandi Press, pp. 9–48, 2008.*