The Effect of Sprint Duration to the Velocity in a Large-scale Embedded Software Project

Dong Seon Park
LG Electronics
Seoul, Korea
dongseon.park@lge.com

Ji Young Noh
LG Electronics
Seoul, Korea
jiyoung.noh@lgepartner.com

Abstract— **Background: Since there are not many empirical studies on the effect of sprint duration on efficiency and productivity in large-scale agile software developments, we collected and analyzed data from large-scale embedded software development. Aim: Our objective is to analyze the effect of sprint duration on the velocity in scrum-based software development projects. Method: We selected five scrums with the least changes from a large-scale embedded software development project of a big company and measured the change in velocity when operating with 3-week and 2-week sprint durations under the same condition. Results: Our analysis revealed that the 2-week sprint duration had about 30 percent higher development velocity. The average velocity was analyzed in four milestones over about two and half years of period, and the one-tailed T-test with an assumption of unequal variances proved that the development cycle of a 2-week sprint duration was higher than that of a 3-week sprint duration. Conclusions: In this respect, this paper contributes to improving the productivity in large-scale embedded software projects.**

Keywords— Agile Development, Sprint Duration, Story Point, Velocity

I. INTRODUCTION

In many engineering fields, such as architecture and systems engineering, projects are traditionally developed using waterfall models. The main characteristics of this approach are distinct requirements analysis, design, implementation, verification, delivery, and maintenance procedures. Each step takes the form of a 'handover' to the next step. Current steps and next steps can provide mutual feedback, but there is no multi-level interaction, which has significant benefits in terms of project predictability. Changing the design or implementation at a later step can require a lot of rework, which can be costly and time-consuming.

Software development has long been developed through traditional project management with the waterfall model, but it is desirable to develop requirements analysis, design, implementation, and verification through frequent iterations to respond to recent industrial diversification, rapid market changes, and customer satisfaction [1].

As shown in Fig. 1, scrum is one of the agile development methodologies and a lightweight framework that helps people, teams, and organizations create value through adaptive solutions to complex problems [2]. The main roles of scrum are product owners, scrum masters, and development teams, and the main processes are product backlogging, sprint planning, sprint backlogging, daily scrum meeting, sprint review, and sprint retrospective.

In scrum model, there's a development iteration called a sprint, which typically lasts between one to five weeks. Although there are various opinions on the sprint duration that can achieve the most efficient results when running scrum, it is generally recommended to run for two or three weeks [3].

In large-scale software development projects involving hundreds of people, the development velocity has a significant impact on product launch schedules, and it is important to optimize duration for predictability. A shorter sprint duration increases agility, but the nature of scrum complicates scheduling due to the overhead of planning and review.

For these reasons, in large-scale scrum projects, changing the duration can be very risky. Therefore, a study of the impact of sprint duration on development velocity in such agile development can provide a basis for scientific decision-making in large-scale agile development projects.

A comparative analysis of development performance and efficiency over sprint duration can provide ideal recommendations for enterprises. This paper aims to analyze the impact of sprint duration on development velocity in large-scale scrum-based software development projects.

II. RELATED WORK

We chose the velocity as the key metric because the burndown chart is widely used to check the work to be done in scrum. Kupiainen et al. [4] studied about metrics in agile and Lean software development. They found that the velocity has the highest influence in occurrences and perceived importance.

A product backlog is a list of features or requirements that stakeholders want to include in a product, and is maintained by the product owner. At the beginning of each sprint, the development team defines the goals and objectives for the sprint and holds daily scrum meetings to monitor progress. At the end of the sprint, developers present the completed features to stakeholders at the sprint review to reflect their feedback into the next sprint backlog. Before starting the next sprint, the development team holds a sprint retrospective to discuss what went well, what needs to be improved and what the team will do next. Sharma et al. [5] found that keeping those ceremonies should be the key responsibility of scrum master to keep track of agile team time spent on agile team ceremonies and agile product build to maximize the agile sprint velocity.

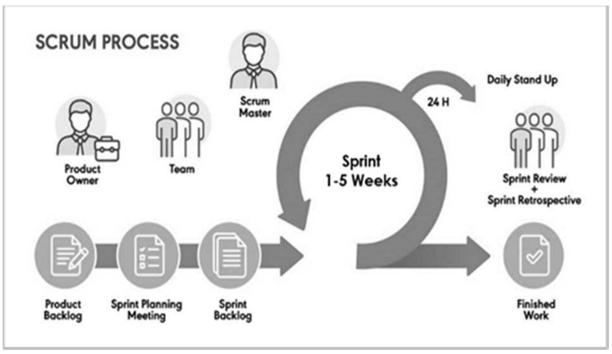

Fig. 1. Scrum Ceremonies and Roles [6]

978-1-6654-5223-6/23/$31.00 ©2023 IEEE

Owais et al. [1] proposed the estimation algorithm using story points divided by the number of resources and days. It can be used when we compare the sprint velocities with various durations.

Al-Sabbagh et al. [7] revealed that the dynamics within software development teams might correlate to their ability to deliver the expected outcome as planned, but not to their ability to develop tasks faster. To control those factors could be very important to get reliable results.

III. METHODOLOGY

To identify and understand the impact of sprint duration on the development velocity, we selected an embedded software development project, which is applied to tens of millions of devices each year.

There had been tens of scrums and they had developed the embedded software including application frameworks, middleware like graphics, and multimedia, kernel BSPs, and so on for several years.

A. Organization

The project involved more than a few hundred developers scattered across the world and operated in more than tens of scrum teams with an average of 5-6 members per scrum. Given the high turnover and the various personnel changes in IT companies these days, we selected five scrums with the least member changes to control the conditions.

B. Development Methodology

The project adopted the scrum of scrums(SoS) model mostly, and all scrums have the same sprint duration with the same milestones.

1) Sprint Duration

In this study, one milestone has a period of 6 to 10 months, and the milestone had a repeated sprint of 3-week or 2-week duration. In Milestones A and B, the project ran a 3-week sprint duration, and in Milestones C and D, the project ran a 2-week sprint duration. All sprint events such as sprint planning, sprint backlogging, daily scrum meeting, sprint review, and sprint retrospective were performed across all four milestones.

2) Story Point

In the scrum model, story point is a unit of measure used to estimate the effort required to complete a task. In this case, one story point is considered to be the amount of work that can be completed by one engineer in one day.

We calculated the total story points to reflect all possible development activities, including not only writing codes but also fixing bugs and making documentation. Fig. 2 shows how we collected and calculated the story points. However, story points do not include sprint ceremonies such as sprint planning, sprint backlogging, daily scrum meeting, sprint review, and sprint retrospective.

3) Velocity

The velocity is the average amount of work performed during each sprint duration in the scrum. The work includes not only writing codes, but also fixing bugs, writing documentations, and so on. To quantify this, the velocity was calculated by dividing the total story points by the number of weeks. For the weekly comparison, we divided the 3-week sprint duration by 3 and the 2-week sprint duration by 2. Since the number of developers participating in each scrum varies, it was also divided according to the number of developers.

C. Overview of the Statistical Analysis

To conduct empirical analysis that is suitable for the purpose of this study, a research model was derived from the previous section and research hypotheses were established.

Compared to the waterfall development methodology, the agile development methodology can respond more quickly, and considering common phenomena in software development such as Parkinson's law [8] and student syndrome [9], it can be hypothesized that the shorter the development duration, the faster the velocity [10].

Thus, the following three null hypotheses were established.

- <Hypothesis 1> Milestones with 3-week sprint durations will have the same velocity.
- <Hypothesis 2> Milestones with 2-week sprint durations will have the same velocity.
- <Hypothesis 3> Milestones with 3-week and 2-week sprint durations will have the same velocity.

The alternative hypothesis to Hypotheses 1 and 2 are the milestones with the same durations can have different velocities. If the milestones with the same duration have the different velocities, hypothesis 3 is meaningless statistically.

The alternative hypothesis to Hypothesis 3 is that milestones with 2-week and 3-week sprint durations will have different velocities, and we can determine which has the faster velocity.

In these hypotheses, milestones refer to a period of repeating multiple sprints for six to ten months. We selected 4 milestones but the duration of each milestone can vary due to the product release schedule. And the sprints can have large deviations due to leaves, holidays, and the number of people, so we compared milestones rather than sprints in this study.

D. Explanation of the Data Collection Process for Scrum Velocity

Table I shows some of the data for each milestone A, B, C, and D tracked over a total of two and a half years. We tried to choose milestones that were as consecutive as possible, and we chose consecutive milestones, B and C, when comparing 3-week to 2-week sprint duration.

To test Hypothesis 1, we collected a sample of nine sprints each from milestones A and B, which had 27 weeks each. We calculated the average story points per week by dividing the number of people per sprint by the number of weeks.

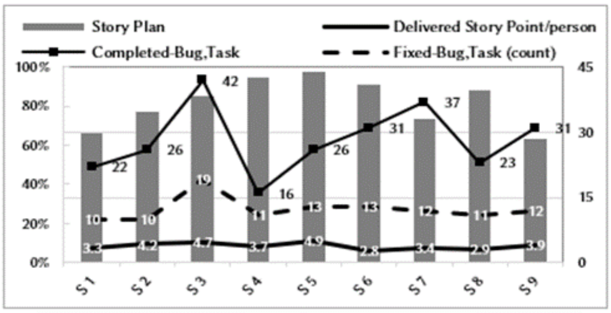

Fig. 2. Story points and average values as an example (Team E of Milestone B)

For the verification of Hypothesis 2, milestones C and D were selected. Milestone C was run for 30 weeks and Milestone D was run for 42 weeks. Since each of these two milestones has a 2-week sprint duration, the average story point was calculated by dividing the number of people and two, respectively.

The milestones B and C were chosen to verify Hypothesis 3, each with 3-week and 2-week sprint durations, and the story points of one week per person were already calculated to verify the above two hypotheses, so we used the story points that were already calculated above. And we also compared the combined data between A and B, and C and D.

Fig. 3 shows the boxplot representing the distribution of the calculated sample data. The boxplot shows that the distribution of the samples of C and D, which had the 2-week sprint duration, are higher than those of A and B where they performed the 3-week sprint duration.

There are some outliers due to various reasons as Pomar et al. found [11].

IV. Results

Hypothesis 1 and Hypothesis 2 assumed equal variances, and Hypothesis 3 assumed heterogeneous variances.

Since the variances or standard deviations of the population is not known for all hypotheses, the hypotheses should be tested with the mean estimated from a given sample, and the number of some samples does not exceed 30, so we used a two-tailed t-test with an assumption of equal variances for Hypothesis 1 and 2, and a one-tailed t-test with and assumption of unequal variances for Hypothesis 3.

Hypothesis 1 – Milestones with a 3-week sprint duration will have the same velocity.

TABLE I. Velocity of each milestone

	A	B	C	D
Sprint 1	1.52	2.02	3.04	1.60
Sprint 2	1.41	2.25	2.12	4.59
…				
Sprint 9	2.46	2.85	2.20	2.20
Sprint 10			2.45	2.09
…				
Sprint 15			3.40	3.18
Sprint 16				3.07
…				
Sprint 20				2.42
Sprint 21				2.22

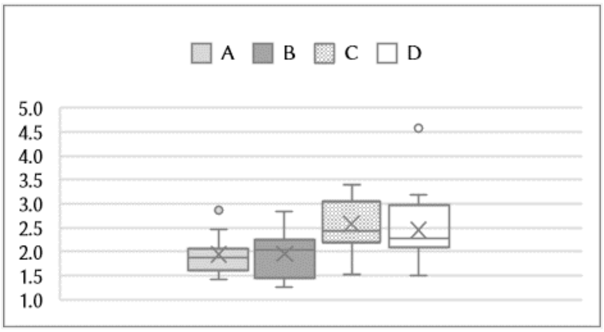

Fig. 3. Boxplot of milestones' calculated story points

The results of the t-test assuming equal variance of the milestones A and B, which performed a 3-week sprint duration, are shown in Table II.

As shown in Table II, the P value of the two-tailed test is 0.97, and the null hypothesis can be accepted at the 95% confidence level, so Hypothesis 1 is acceptable. Thus, it can be said that milestones with a 3-week sprint duration have the same velocity.

Hypothesis 2 - Milestones with a 2-week sprint duration will have the same velocity.

The results of the t-test assuming equal variance of the milestones C and D, which performed a 2-week sprint duration, are shown in Table III.

As shown in Table III, the P value of the two-tailed test is 0.56, and the null hypothesis can be accepted at the 95% confidence level, so Hypothesis 2 is acceptable. Thus, it can be said that milestones with a 2-week sprint duration have the same velocity.

Hypothesis 3 - Milestones with 3-week and 2-week sprint durations will have the same velocity.

The expectation in this study for Hypothesis 3 is that the null hypothesis is rejected, and the alternative hypothesis is accepted. In other words, 3-week and 2-week sprint durations will not have the same velocity. We will also verify whether 2-week sprint durations have a faster velocity using a one-tailed test.

Since Hypothesis 1 and Hypothesis 2 have already shown that they have the same velocity, the comparison between B and C can be representative of the sample.

Table IV shows the results of the one-tailed t-test of the two consecutive milestones in the sample in Fig. 3: milestone B, a 3-week sprint duration, and milestone C, a 2-week sprint duration.

TABLE II. t-test of Hypothesis 1

T-Test : Two-Sample Assuming Equal Variances	A	B
Mean	1.93888889	1.94666667
Variance	0.22458611	0.26355
Observations	9	9
Pooled Variance	0.24406806	
Hypothesized Mean Difference	0	
df	16	
t Stat	-0.0333969	
P(T<=t) two-tail	**0.9737712**	
t Critical two-tail	2.1199053	

TABLE III. t-test of Hypothesis 2

T-Test : Two-Sample Assuming Equal Variances	C	D
Mean	2.57533333	2.44761905
Variance	0.28246952	0.51724905
Observations	15	21
Pooled Variance	0.42057513	
Hypothesized Mean Difference	0	
df	34	
t Stat	0.58253458	
P(T<=t) two-tail	**0.5640503**	
t Critical two-tail	2.03224451	

TABLE IV. T-TEST OF HYPOTHESIS 3

T-Test : Two-Sample Assuming Unequal Variances	B	C
Mean	1.94666667	2.57533333
Variance	0.26355	0.28246952
Observations	9	15
Hypothesized Mean Difference	0	
df	17	
t Stat	-2.8660373	
P(T<=t) one-tail	**0.0053526**	
t Critical one-tail	1.73960673	
P(T<=t) two-tail	**0.0107052**	
t Critical two-tail	2.10981558	

As shown in Table IV, the P value is 0.01 in the two-tailed test, and the null hypothesis is rejected at a 95% confidence level. And the P value in the one-tailed test is 0.005, so it can be said that the 2-week sprint duration has the faster velocity than the 3-week sprint duration. Moreover, the average weekly development velocity of 3-week sprint durations is 1.95, and the average weekly development velocity of 2-week sprint duration is 2.58. The latter is about 30 percent higher.

Table V shows the t-test of combined milestones A and B, and C and D. The one-tailed p-value is 0.0004, so the comparison of combined milestones A and B, and C and D is also validated to have a faster velocity for 2-week sprint durations, which is the same result as the single milestone B and C comparison. In this case, the average velocity for the 2-week sprint duration is 2.50 and the average velocity for the 3-week sprint duration is 1.94, meaning that the milestones in the 2-week sprint duration are about 29% faster.

The higher average velocity for 2-week sprint durations than 3-week sprint durations is significant despite the higher overhead of sprint ceremonies. It also confirms the findings of Sharma et al. [6] that better adherence to sprint ceremonies leads to faster development velocity.

V. DISCUSSION

Theoretical Implications

The results of this study provide implications that short development durations can lead to higher development productivity in terms of development efficiency taken by scrum models in large-scale software development. This is consistent with the fundamental purpose of Agile, and despite the various events of the scrum model, the 2-week sprint duration can sufficiently guarantee the velocity.

Practical Implications

Applying the results of this study to the performance of the development duration of large-scale software development organizations, the application can be considered in practice as follows.

TABLE V. T-TEST OF COMBINED MILESTONES

T-Test : Two-Sample Assuming Unequal Variances	A,B	C,D
Mean	1.94277778	2.50083333
Variance	0.22972712	0.41263643
Observations	18	36
Hypothesized Mean Difference	0	
df	44	
t Stat	-3.5854846	
P(T<=t) one-tail	**0.0004193**	
t Critical one-tail	1.68022998	
P(T<=t) two-tail	**0.0008386**	
t Critical two-tail	2.01536757	

Changing the sprint duration in large-scale software development is a very important decision that can affect the timing of mass production, so if implemented without sufficient scientific verification, it can act as a major risk factor. Therefore, scrum development organizations that currently have longer than 3-week sprint duration can expect to improve velocity productivity by reducing the sprint duration as 2-week based on this study.

VI. FUTURE WORK

A. *Further Research on the Impact of Sprint Durations on Other Agile Models*

Although it was a large-scale software development project, selecting only 5 scrums, which has the least changes in the scrum among hundreds of people may have limited the study. However, it is considered a statistically significant method as it is increasingly rare to continue to develop the same module due to frequent turnover in the IT industry and voluntary work conversion to gain various experiences.

We may study 1-week sprint durations instead of 2-week to see the results, but as we will see in the next section, 2-week sprint durations have increased developer fatigue, so a comprehensive study including other growth factors of developers is needed rather than just thinking about shortening the duration to increase the velocity.

In this study, statistical validation was conducted on a sample of five scrums in the Scrum of Scrums (SoS) method, which is an extension of Scrum. However, it would be interesting to study the impact of sprint durations on development efficiency in Scaled Agile Framework(SAFe) or Large-Scale Scrum(LeSS), which are more widely used among large-scale agile development frameworks.

B. *Discussion of the Retrospective Survey Results*

After switching the sprint duration from 3-week to 2-week, a survey was conducted on developers for the milestone retrospective. In the survey, many developers expressed fatigue as a result of the shortened sprint duration. Therefore, it would be necessary to verify whether there is a slowdown in velocity due to the fatigue of the developer and whether Test Driven Development is carried out well when it continues 2-week sprint durations for many years.

Also, currently all scrums are developing with the same development cycle. However, as proposed by Anand et al. [12], it will also be meaningful to see the development impact when each scrum has its own development cycle that matches that scrum. However, when using this method, it is essential to have a process that can control the entire development to be carried out with the same rhythm overall.

VII. CONCLUSION

In the scrum model, it is very important to set the sprint duration. Taking a very short duration can lower velocity by shortening the time to develop due to the overhead of ceremonies like sprint planning, sprint review, and sprint retrospective. On the other hand, if the sprint duration is too long, agility, which is the fundamental purpose of Agile, can be decreased.

This study explored the impact of the sprint durations on development efficiency and productivity in agile software development, which is emerging these days, and empirically collected and analyzed data from a large-scale embedded software development project.

In summary, it was found that shorter sprint durations can speed up development. The average velocity was analyzed in four milestones over a total of two and half years, and the one-tailed T-test with an assumption of unequal variances proved that the velocity of 2-week sprint durations was about 30% higher than that of 3-week sprint durations.

REFERENCES

[1] M. Owais and R. Ramakishore, "Effort, duration and cost estimation in agile software development," 2016 Ninth International Conference on Contemporary Computing (IC3), Noida, India, 2016, pp. 1-5, doi: 10.1109/IC3.2016.7880216.

[2] J. Sutherland and K. Schwaber, “The scrum guide. In: The definitive guide to scrum: The rules of the game,” 2020. [Online]. Available: SCRUM.org.

[3] S. Dhir, D. Kumar, and V. B. Singh, “An estimation technique in agile archetype using story points and function point analysis,” Int. J. Process Manag. Benchmarking, 2017, doi: 10.1504/IJPMB.2017.086933.

[4] E. Kupiainen, M.V. Mäntylä and J. Itkonen, “Using metrics in agile and Lean software development - A systematic literature review of industrial studies,” Information and Software Technology 62. 2015, pp. 143-163.

[5] S. Sharma, D. Kumar and M. E. Fayad, "An impact assessment of agile ceremonies on sprint velocity under agile software development," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596508.

[6] The Agile Journey: A scrum overview, 2023, https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/

[7] K. Al-Sabbagh and L. Gren, "The connections between group maturity, software development velocity, and planning effectiveness," Journal of Software: Evolution and Process 30, 1 (2018),e1896

[8] C. Parkinson, "Parkinson's law," Economist. 1955: https://www.economist.com/news/1955/11/19/parkinsons-law

[9] D. Smith, "The effects of student syndrome, stress, and slack on information systems development projects," Informing Science and Information Technology. 7, 2015, pp. 489–494. doi.org/10.28945/1222.

[10] T. DeMarco and T. Lister, “Waltzing with Bears: Managing risk on software projects,” July 2013, Addison-Wesley Professional

[11] F. Albero Pomar, J. A. Calvo-Manzano, E. Caballero, and M. Arcilla-Cobián,“Understanding sprint velocity fluctuations for improved project plans with scrum: a case study,” J. Softw. Evol. Process, vol. 26, no. 9, pp. 776–783, 2014, doi: 10.1002/smr.1661.

[12] A. Anand, J. Kaur, O. Singh and O. Alhazmi, “Optimal sprint length determination for agile-based software development,” Jan. 2021, Computers, Materials and Continua, 62, 37-65.