Spikes in Agile Software Development: An Empirical Study

Hussein Al Hashimi
Electronics and Computer Science
University of Southampton,
Southampton, UK
King Saud University, Riyadh, KSA
hah2n17@soton.ac.uk

Andy Gravell
Electronics and Computer Science
University of Southampton,
Southampton, UK
amg@ecs.soton.ac.uk

***Abstract*— Spikes can be an essential component of the agile development cycle, because they assist the teams, for both technical and functional issues, to identify any uncertainty in a user story, leading to a more efficient solution to the problem. The use of spikes in agile software development (ASD) can enable organizations to produce quality software by employing the required technical expertise, planning the entire development cycle and ensuring that the client's requirements are adhered to. This study aims to examine the use of spikes in ASD. It explores the role, efficiency and efficacy of spikes in various software development domains through the different agile methods. An exploratory research design is adopted to achieve this purpose, whereby mixed methods are used to collect concurrently both qualitative and quantitative data from the experts recruited to the study. Through the survey, it establishes that the primary role of spikes is risk management through investigations to understand user stories and reveal any uncertainty. Conclusively, the study findings imply that spikes have become an essential tool for most agile teams in ASD. The efficiency and effectiveness that are reported show that the majority of experts in software development have realized the value of using spikes in their processes.**

Keywords-Agile, spikes, prototype, risk management, uncertainty.

I. Introduction

Software development is an expanding and evolving concept. Today, many software projects need to incorporate changing customer requirements, which makes them difficult to deliver without a concise and well-thought-out approach. Old methodologies can no longer satisfy market needs and, today, software processes are considered more flexible to meet the productivity, efficiency and flexibility goals [1]. In changing environments, the agile software development approach is more adaptive to cater to the customer's needs, values and principles and to current practice in modelling software to be applied effectively in a software development project. A concept known as eXtreme Programming (XP) has been thrust into the agile methodology limelight, offering the required values, principles and practices for today's software development projects [2]. One practice inherent in XP is the spike solution, a concept that attempts to minimize risk and to improve customer availability and unit testing, while meeting coding standards and achieving optimization [3].

Spikes can be defined as a particular type of story involving activities such as research, investigation, exploration, prototyping and design, with the aim of reducing or driving out the uncertainty or technical risks associated with either the user story or other facets of the project [4]. In addition, spikes in agile determine the uncertainty of a project by collecting relevant and required pieces of information to help in understanding its technical or functional requirements. For instance, spikes are used when an agile team needs to resolve a specific technical problem or does not have enough information to estimate a user story [5]. Spikes are treated in the same manner as user stories, adopting the standard format: goal; persona; and benefit. Their major advantage is inclusion of the research goal in place of the direct user value. Because spikes do not deliver direct value, they run no risk of slipping back into the behaviour of the waterfall method [6].

The primary goal of this study is to investigate three research questions relating, respectively, to spikes' roles, efficiency and efficacy in risk management. Furthermore, it aims to depict aspects of spikes in various development domains. This article highlights spikes' usage in agile development processes, covering several techniques and approaches. It addresses the basic concepts of spikes' implementation in agile methodologies by considering their use in risk management.

II. Background and Related Work

Within agile software development, investigation or feasibility typically occurs where information is required as part of an investigation or research, this is known as a spike or a spike solution. Consider spike as an unknown quantity for the full agile team, where the investigatory element of the agile development is where unknown or uncertain elements need to be learned or identified e.g. through a user story, or a new technology to provide an accurate estimation [7]. A formal definition identified from Cohn (2005) defines a spike as an iteration task undertaken to gain a better understanding for basic knowledge or to answer a question.

Leffingwell (2010) describes spikes as a special story that drives out risks and uncertainty within a user story, specifically where knowledge is light or where a spike can be used to form the base of research to mitigate risk. Miranda (2009), further describes a spike as an experiment to attempt to learn or seek further knowledge on something. Spikes are essential in Agile to assist in mitigating risks and issues within a technical and functional setting by gaining an understanding of a particular subject area, through understanding of requirements or enhancing accuracy of a story [4].

The origination of spikes came from eXtreme Programming (XP), it is particularly applied in small teams comprising less than ten developers. Customers are normally part of the team as they help in the approval of the process after meeting the end user's needs. In XP, the role of a spike is to get access to information required to mitigate risks of this technology-based approach and validate estimates [10]. According to Leffingwell, XP-originated user story was introduced and adopted as the primary currency for expressing application requirements in agile development practices.

978-1-7281-7624-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCI51800.2020.00319

Architectural spikes are used in XP and are iterations that demonstrate certain technological approaches [10]. When the information is identified, this is shared to all project stakeholders [10] so everyone associated with the project understands the particular area before it is released into Release Planning. When released back into the Release Planning stage, the “architectural code” and user stories are considered understood and the project team produces a release schedule to divide the project into iterations, where the design and development of the project begin to implement the code. Within the iteration stage, along with the user story documentation and code designs, the latest version of the design is released into acceptance testing as shown in Figure 1 below.

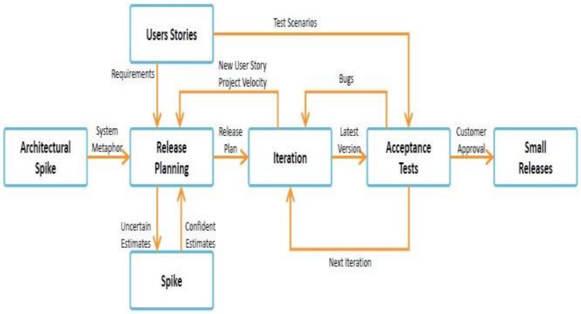

Fig. 1. Spikes in eXtreme Programming

Bugs, or formally known as defects, would be identified as part of the acceptance testing and returned for resolution within the Iteration stage until all bugs are resolved. User stories are used to define the testing scenarios to ensure all functional areas defined in the user stories are captured within testing coverage. Where it is identified that a new user story is required from within the Iteration, the amount of work required is measured within a framework. This is known as Project Velocity, calculated at the end of the Iteration to predict the number of user points they should plan in order to complete the iteration [8].

III. Research Methodology

The research methodology is a systematic plan for undertaking research. This section mainly focuses on how the research questions will be answered. The research methods undertaken in this study are both quantitative and qualitative. Mitchell (2018) states that mixed methods should be considered, as they are attractive to new researchers who have an open mind and wish to solve a particular problem. In this case, we want to assess the efficiency and efficacy of spikes in risk management through the different roles of spikes. To achieve that, concurrent mixed-methods has been employed which include a combination of quantitative and qualitative data collection techniques.

The initial step in any study involves research questions that are focused on an issue or a specific concern. This establishes the idea of the study and the gaps to be filled [12]. As derived from the literature review [4], this study’s research questions can be expressed as follows:

- RQ1. What roles do spikes play in different agile methods?
- RQ2. How can agile spikes be used efficiently?
- RQ3. How can agile spikes be used to manage risk effectively?

In this study, a questionnaire was used to answer the research questions. The questionnaire approach involved experts with experience in both spikes and ASD. As part of the data collection instrument, the questionnaire has 30 questions, mostly comprising closed-ended questions but with a few open-ended questions. The criteria for inclusion in the questionnaire considered the convenience and willingness of the invited participants.

Questionnaire design largely depends on the objectives or the research questions to be addressed. The questions need directly and collectively to address adequately the goals of the research. Without a proper design, a questionnaire may collect redundant or irrelevant information, which may change the direction of the study or impede the fulfilment of the objectives altogether [13]. Although there are instances where standardized questionnaires are employed unless it is a replication every study requires an entirely new questionnaire.

In this study, the questionnaire was based on the focus of the research. Thirty questions were formulated: demographic details; the efficiency of spikes in ASD; and the efficacy of spikes in risk management. Each of these sections, mainly the last two, contained a mix of closed- and open-ended questions. Together, they answered RQ1, RQ2 and RQ3. Thirteen of the total required a Likert-scale response, being statements that prompted the experts to either agree or disagree on a five-point scale.

Estimates of the necessary sample size should be made to establish the minimum number of participants to achieve a specific power for the results in the tests being run [14]. While there are several formulae for calculating sample size, in most cases the population size is unknown. Hence, the size has to be based on other parameters, such confidence level and an attribute’s estimated proportion [15].

Since the target participants are experts in ASD and spikes across the globe, it is not possible to establish the total number of professionals in this field. Therefore, the G*Power tool was used to calculate the minimum estimated sample size: given that the researcher expected power of 0.95, an effect size of 0.5 when a t-test is a statistical test to be used during analysis. According to Cohen [16], sample size estimation using G*Power depends on the effect size that is established; however, the estimate used a hypothesized effect size of 0.5 to obtain a sample size of 54 participants. Due to the need to clean the data and eliminate outliers, the larger sample size was used. In the end, after eliminating non-responders and those who recorded no experience in either ASD or spikes a sample of 72 participants was obtained.

In research methods, there are two types of sampling: probabilistic and non-probabilistic. The latter refers to sampling techniques in which not all individuals in the population have an equal chance of being included in the study sample. In contrast, the former refers to random sampling criteria under which everyone has an equal chance of inclusion [17]. This study used a non-probabilistic sampling, and only those invited by the researcher could access the questionnaire.

IV. Results

This section describes the results of the survey. The questionnaire was devised to collect data, and software development experts were invited to complete it. It primarily addresses RQ1, RQ2 and RQ3 by assessing the roles, efficiency and effectiveness of spikes in ASD.

The questionnaire consists of 30 questions, comprising both closed- and open-ended questions. The link to the questionnaire was made available to participants carefully picked from the researcher's professional relationships, as well as social networks. Of the total targeted, 83 consented to participate in the survey and its link was provided to them; however, after data cleansing to eliminate some random responses, data from only 72 participants were used in this research. The cleaning involved discarding participants who made random or non-responses to critical questions, including on agile roles, and those who had no experience in spikes. In the data collected, there were no outliers since the quantitative questions used a predetermined scale. Those experts included in the final sample had at least one year of experience and completed all the questions in the questionnaire.

As mentioned earlier, data cleansing involved excluding participants who had no experience of using spikes or agile development. Therefore, all 72 participants have either used or witnessed spikes' use in ASD. The respondents' average experience with agile developments was found to be 6.69 years, with a standard deviation of 3.1 years. In the same context, their experience with spikes had an average of 5.19 years and a standard deviation of 3.2. This implies that, generally, the experts have more experience of agile development than of spikes. It also suggests that the application of spikes in agile development is adopted after an expert gains experience in using ASD strategies. A summary of participants' responses is represented in Figure 2 below.

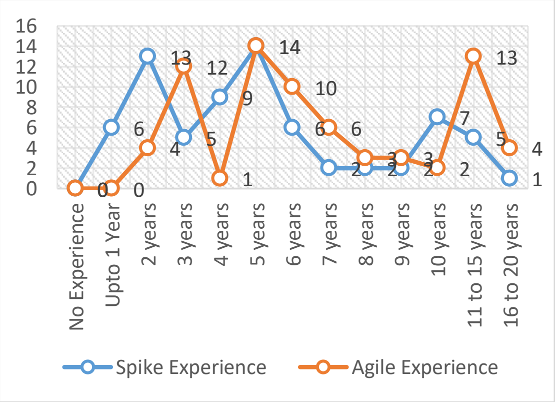

Fig. 2. Participants' experience of using spikes in agile software development

After cross-tabulating the agile roles and experiences in ASD and spikes, it was found that most participants had experience of one to five years. About 49 participants had between one to five years of experience in ASD, and 81 had the same in utilizing spikes in ASD. Similarly, 48 had experience of between six and 10 years in ASD, and 47 had the same experience with spikes in ASD. Notably, only 10 participants had experience of between 16 and 20 years in ASD and only two respondents had this in spikes in ASD.

Of the participants, there were 63 Scrum masters, 18 product owners, eight developers, eight testers (QA), 40 Agile Coaches, two Project Managers (PM), two Business Analysts (BA) and one Subject Matter Experts (SME), as shown in Figure 3 below. Some participants had more than one role, making the tally more than the sample size of 72.

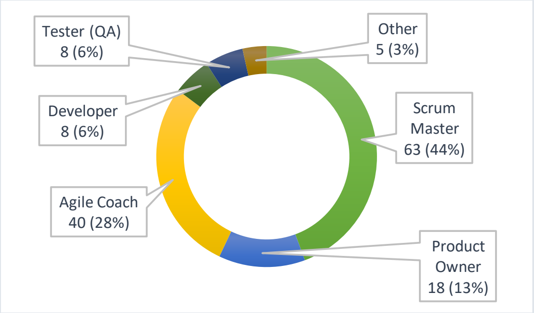

Fig. 3. Participants' agile roles

On assessing participants' experience of spikes and agile developments, a positive relationship was established. As seen in Figure 4, the relationship showed a strong (almost perfect) relationship between the two variables. Based on the plot, the greater the experience that a person has in agile development, the greater their experience in the use of spikes in ASD. Therefore, it can be concluded that when individuals consistently become accustomed to ASD, their need for applying spikes increases, elevating its usage in risk management and the estimation of user stories during the ASD process.

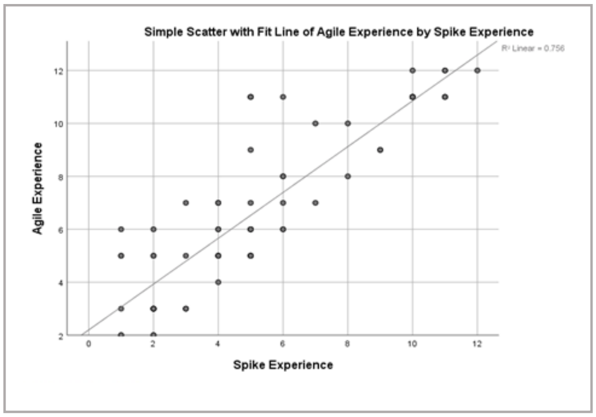

Fig. 4. Scatter plot of spike experience vs. agile experience

A. Roles of Spikes in Agile Software Development Methods

This section answers RQ1, which seeks to find out the specific roles that spikes play in various agile methods.

In evaluating the agile methodologies used, the questionnaire required participants to select/state the agile method that they employ mostly. The most common was found to be Scrum, at 90.28%, and the least Dynamic System Development Method (DSDM), at 1.39% of participants. The Scrum method's dominance explains why most participants are Scrum masters and have the most experience in this role. As shown in Figure 5 below, other responses accounted for 1.39% of participants. This response points to a mixture of practices as a method utilized in agile development.

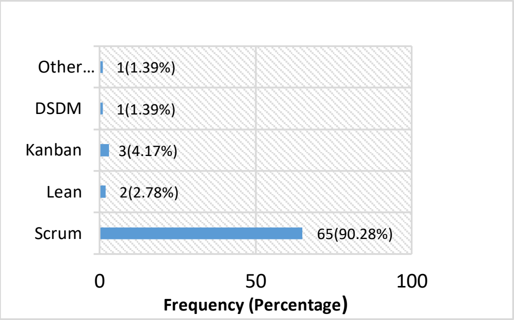

Fig. 5. Most-used agile methods

Although most of the participants stated different roles of spikes, they agreed on the following functions:

- Helping in better understanding requirements
- Driving a risk-aware culture, embedding more collaboration with accountability
- Helping in estimation and risk management
- Exposing complexity/simplification
- Providing familiarization among team members
- Experimenting with an approach to the delivery of a specific product or story to the client
- Solving technical debts.

In stating the roles of spikes, in total 66 participants agreed that they are used in prototyping, exploration, investigation, and design and research activities. Only four participants refuted this claim, while two were neutral.

Responses provided by participants included: 'It allows the team member to explore new ideas, sparks creativity when used properly, but of course it depends how much liberty team has on such things.' Additionally, participants argued: 'Spikes can improve developers' domain and architecture knowledge and make better connections among team members. According to one participant, spikes create an opportunity for team members to grow together. In particular, the expert said: 'when team members work on the same spikes in parallel and when spikes are properly documented, they can give an opportunity to team members to grow together, have better feature insight and learn how to improve their cooperation and communication.'

B. *Effectiveness of Spikes in Agile Software Development*

In regard to the efficiency of spikes, the participants were answered the questions related to the RQ2. In response, 54.2% of participants, agreed that spikes are efficient in ASD, and 30.6% strongly agreed with the same question. However, about 15% disagreed or were neutral about the effectiveness of spikes in agile development.

In regard to the efficiency of spikes in improving software products' quality, most respondents, at 91.67% (66), confirmed that spikes are useful. Although 6.94% (5 participants) denied that their application does help, the majority responded in affirmative, providing further evidence of the usefulness of spikes in risk management in ASD. Nevertheless, about 1.39% of respondents were unsure about the effectiveness of spikes in improving the quality of end products of ASD.

From the responses, 18 experts strongly agreed that the efficiency of spikes in ASD depends on the spikes applied, while 30 agreed to the same. However, 13 disagreed and four strongly disagreed. In the same context, 26 strongly agreed that efficiency depends on the team applying the spike, while 29 agreed; six disagreed and three strongly disagreed.

There are two types of spikes that this study seeks to explore. Technical spikes are the most common in ASD. Of the 72 participants, 79.17% asserted that technical spikes are the most commonly used, while 20.83% said that functional spikes are more common.

In terms of efficiency, half the respondents affirmed that both technical and functional spikes are efficient when utilized in ASD. However, 43.1% claimed that technical spikes were the most effective type when applied in agile development, as shown in Figure 6 below.

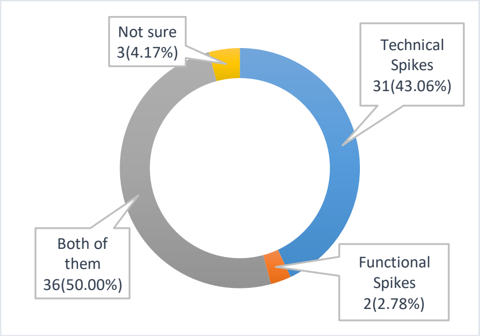

Fig. 6. Efficiency of spike types

The results show an increased use of technical spikes in agile development in various domains. Although most believe that both types are efficient when used, technical spikes are more commonly used, and the response best represents it in the various domains.

The application of spikes in ASD is known in multiple domains, using various methods. According to the data obtained, user experience design (UX), at 22.73% (35 responses), and cloud computing, at 15.58% (24 responses), lead in the domains that utilize spikes frequently. Computer science education is the domain that least uses spikes, at only 2.60% (4 responses) of the respondents affirming it, as shown in Figure 7 below.

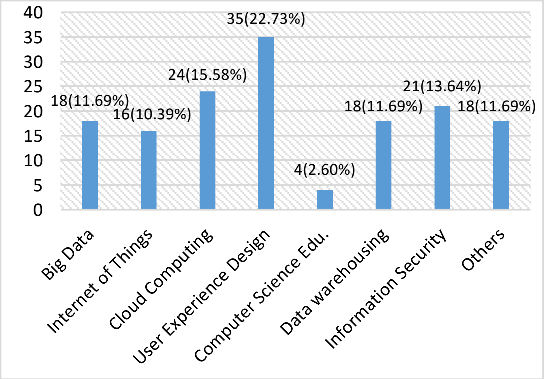

Fig. 7. Efficiency Domains in which experts apply spikes

It is evident that the other domains such as e-commerce, SaaS, web development, mobile apps and healthcare software are critical areas in which spikes are extensively applied. Most participants reported using spikes or having seen them used by others in more than one domain, hence a large number of responses, exceeding the sample size.

According to the responses, 34.72% (25 experts) feel that it is somewhat likely that spikes are applied to estimate user stories in ASD projects. Cumulatively, 54.16% (39) mentioned that it is likely that spikes are applied in estimating user stories. However, 20.84% (15) argued that it is very unlikely for spikes to be applied specifically to estimate user stories. About 25% (18 responses) were unsure if experts are likely to use spikes in estimating user stories as seen in figure 8 below.

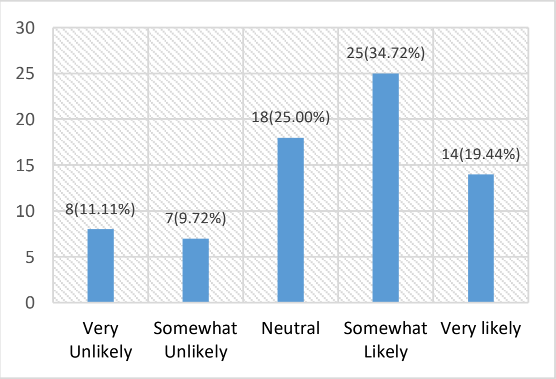

Fig. 8. Likelihood of experts applying spikes in estimating user stories

C. *Managing Risk with Spikes in Agile Software Development*

In this section, RQ3 will be answered through the results. Due to spikes efficacy, they were reported to be used frequently in ASD to minimize risk. According to the data collected, 36 of respondents (50%) confirmed that they often used spikes in their roles in agile development to minimize risks. Nine (12.5%) asserted that they always used spikes or had seen them used in ASD. Only five (6.94%) said that they rarely used spikes, while 22 (30.56%) stated that they used them sometimes.

Based on Q17, a total of 33 experts agreed that spikes are the best way to mitigate risk. Only 10 disagreed, while 29 were neutral on the issue. In Q19, all participants apart from four, who were neutral about the statement, agreed that spikes are effective in reducing uncertainty in ASD. More than half of the participants, 47, agreed that spikes can estimate user stories more precisely in ASD projects (Q20). Only seven disagreed, while the rest were neutral. In Q21, a total of 36 experts agreed that spikes should be used sparingly, and 25 disagreed: the other 11 were neutral. A total of 61 agreed that spikes can be used when there is uncertainty about a process, and only five respondents disagreed.

The final Likert-scale question, Q26, sought the opinions of participants on whether risks in ASD can be managed without the application of spikes. They were asked to agree or disagree with the statement and provide their rationale for the response. Based on the answers provided, most participants agreed with the statement and only 24 disagreed. Those who agreed that risks could be managed without using spikes gave various reasons. One of the participants mentioned: ‘When the risk is related to release deadline, we do not have enough time for spikes and having a lot of spikes can even enhance the risk by slowing down the development process and delaying the deployment to the client. On the other side, when we have a flexible deadline, and we can focus on quality, spikes can mitigate or even eliminate risks in the software development process.’

An expert with a similar opinion on spikes and risk management in ASD asserted: ‘If a product owner/stakeholder does all the necessary research before writing the user story then there is no need for the development staff to use spikes to be able to estimate those stories because there should be no outstanding questions.’

Other responses included as rationales for agreeing with the statements were:

- Spikes are not the only way of mitigating risks in ASD.
- It is possible to push story point estimates higher to capture risk and leads to fewer stories being taken into a sprint.
- Sometimes there is limited time to apply spikes, and too many can slow the development process, increasing the risk.
- Better application of Scrum in agile environment can make spikes unnecessary.

Although most agreed that projects could be completed without applying spikes, others disagreed. In particular, one participant mentioned: ‘Without a spike, the team leaves themselves exposed to prolonging the time spent on a specific story without getting close to a solution.’ A different stated: ‘Sometimes, no matter how experienced the team is or how easy the project is to implement, there are requirements for which a technical solution is not easy to decide. In this case, a spike is an easy way to overcome the problem.’ Lastly, a participant mentioned how a ‘spike is the best option to take time to work on identified risk’.

The efficacy of spikes in ASD different development domains, and specifically in reducing risk, was measured on a five-point Likert scale. The experts were asked to score the effectiveness of spikes in ASD. A score of 5 represented ‘Very effective’, 4 ‘Partially effective’, 3 ‘Neutral’, 2 ‘Not effective’, and 1 ‘Completely ineffective’ as shown in Figure 9 below.

5
Very Effective
4
Partially Effective
3
Neutral
2
Not Effective
1
Completely Ineffective

Fig. 9. Likert rating scale for the effectiveness of spikes in ASD

From the responses, the efficiency of spikes in the domains is as presented in Table 1. About 56.94% of experts scored the efficiency of spikes in the domains that they mentioned at 4 of 5, and only 13 participants rated it below 4; that is, 12 experts rated it 3/5 and just one rated it 2/5. Overall, 81.94% (59 participants) scored the efficiency of spikes in domains above 4/5.

In the same manner, participants were asked to rate the efficiency of using spikes in ASD. Around 48.61% of the participants scored it 4/5 while 31.94% scored it 5/5. The percentages represent 35 and 23 participants, respectively. However, 19.44% (14 responses) scored the efficiency at 3/5.

To ascertain the efficacy of the application of spikes in risk management, the participants were asked to rate it out of 5 possible points. The greatest proportion of experts, 54.17%, rated it 4 out of 5 and 19.44% (14 responses) scored it 5/5. Only one expert scored it as 2/5 regarding risk management, as shown in Table 1 below.

TABLE I. SUMMARY DISTRIBUTION OF SCORES FOR SPIKES EFFICIENCIES AND EFFECTIVENESS

Measured item	Scores	Frequency	Percentage
Efficiency of spikes in different domains	5	18	25.00%
	4	41	56.94%
	3	12	16.67%
	2	1	1.39%
Efficiency of spikes in ASD	5	23	31.94%
	4	35	48.61%
	3	14	19.44%
Effectiveness of spikes in risk management	5	14	19.44%
	4	39	54.17%
	3	18	25.00%
	2	1	1.39%

On average, the respondents scored spikes' effectiveness in various domains at 4.06, with a standard deviation of 0.68, while their efficiency in ASD scored 4.13 on average with a standard deviation of 0.71. Similarly, spikes' efficacy in risk management scored on average 3.92 with a standard deviation of 0.70. As shown in Figure 10 below, the median response for all three variables was found to be approximately 4.0. This value also represents the computed mean response of the scores for each variable. The upper cut-off (maximum score) across all the three variables was 5.0; however, the lower cut-off (minimum score) was 2.0 for the efficiency of spikes in domains, 3.0 for the efficiency of spikes in ASD and 2.0 for the effectiveness of spikes in risk management. The lower cut-off for spikes' efficiency in domains was treated as an extreme outlier in the data, despite falling in the range of 1 to 5, as shown in Figure 10.

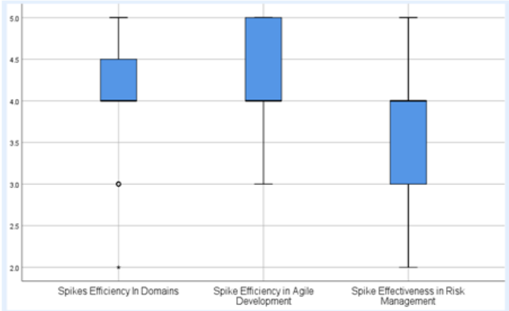

Fig. 10. 'Box and whisker' plot for efficiency and effectiveness scores of spikes

V. Discussion

The application of spikes in ASD continues to be embraced by most experts in this field. Spikes' usefulness, efficiency and effectiveness are core reasons why professionals in software development find them suitable. As the experts pointed out in the questionnaire, the roles of spikes in estimating user stories, managing risk and researching are the primary reasons why they are used in software development projects. On average, participants had an experience of over six years in ASD and over five years in using spikes in their various agile roles. The information that they provided is relative to their experience in the field. The results of the questionnaire addressed RQ1, RQ2 and RQ3 in detail, providing both quantitative statistics and qualitative information to answer the research questions succinctly.

The questionnaire was designed to ask open-ended questions to gather more information on the roles that spikes play in ASD. As reported in the results, most roles mentioned are similar to those cited by the experts interviewed in this study. The key roles revealed in the questionnaire centred on estimation, investigation, experimentation and risk mitigation. In particular, spikes are said to be essential in prototyping and investigating the dynamics of a new system or technology that is being applied in a project. Whenever a team intends to solve technical issues in a software development project, the application of spikes is necessary to allow agile teams to prototype and gain more information on a technological approach, enabling them to subject a user story to some future time-box [5].

Spikes are known to be useful in providing familiarization with new systems, as well as solving a technical debt. In ASD, technical debt is created when a certain task is delayed so that developers can meet a particular deadline, to obtain a short-term benefit [18]. Through an agile spike, technical debt can be resolved by enabling a team to estimate the effort and time needed to complete each activity, allowing them to plan the development process and minimize the occurrence of technical debts. In regard to familiarization, spikes are useful in investigating the system design to understand how the new technology operates. According to Leffingwell (2010), spikes may be used to undertake basic research on systems, technology or domains in order to familiarize the developers. This is regardless of the methodology being used or the domain in which the spikes belong.

Other roles mentioned by the experts include driving a risk-aware culture in an agile team and allowing its members a common understanding of the user stories. In terms of estimation, spikes estimate not only user stories but the risk and complexity of the software development project. The same spikes are used to resolve the complexities, whether functional or technical. In general, spikes play a significant role in ensuring that an ASD project faces few risks and disruptions due to unclear user stories or user requirements [19]. Although there are experts who do not perceive spikes to be the best technique for estimation in ASD, they maintain that when they are applied properly they are indeed useful.

Spikes have also been found to be efficient in ASD. The application of this agile technique is valuable not only to the teams but to the products developed. According to the participants in the questionnaire, using spikes has been associated with improved quality of the software produced at the end. In the questionnaire, they scored an average of 4.13/5 in ASD for spikes' efficiency. The high score is indicative of the positive implication that their application has for software development projects. The efficiency is more on the ability of spikes in the estimation, prototyping, risk management and exploration of uncertainty in a software development project.

Risk management is emerging as an inevitable responsibility that experts in software development need to undertake. Of the many ways of executing this responsibility, the use of spikes is proving to be the best to break the deadlock when risks are created in ASD projects. Since spikes have been found to be efficient in various domains and ASD in general, their effectiveness in risk management has been reported to be high.

From Q19 in the questionnaire, it is apparent that most experts consider spikes to be effective in resolving uncertainty in software management projects. However, the effectiveness is determined by accurate identification of the potential risks to be assessed and controlled. According to Albadarneh and Quesef (2015), in order to know what action is to be taken and planned to reduce the risk, we first have to identify it through analysis of the architectural design of the software. In this regard, it is valid to conclude that spikes are not only efficient but effective in managing risks in ASD projects.

VI. Conclusion

This paper focused on answering the three research questions through a survey of 72 experts. The results include both quantitative and qualitative data from the 30-item questionnaire, and they show that spikes are generally prominent in ASD projects. As the experts had various roles and substantial experience in both agile and using spikes, they contributed significant information that proved that the efficiencies of spikes in various domains and ASD projects are positively correlated to their effectiveness in risk management.

In essence, when the efficiency of spikes is increased, the technique becomes more effective in addressing any risks that arise in agile software projects. Furthermore, the application of spikes was reported to be essential in undertaking estimates of both user stories and the effort required to complete iterations, hence the whole project. The information provided throughout this paper further answers the three research questions and makes reference to empirical evidence obtained from the quantitative data collected. Conclusively, the study provides a detailed and evidence-based account of how spikes are used efficiently in ASD to estimate user stories, reduce risk and improve the overall quality of the software project being developed.

References

[1] M. Hneif, SH. Ow, "Review of agile methodologies in software development," International Journal of Research and Reviews in Applied Sciences, 1(1), pp.1-8, 2009.

[2] K. Beck, C. Andres, Extreme Programming Explained: Embrace Change. Addison Wesley Professional, Boston, 2004.

[3] D. Karlström, "Introducing extreme programming-an experience report". In Proceedings of the 3rd international conference on eXtreme processing and agile processing software engineering, 2002.

[4] H. Al Hashimi, A. Gravell, "A Critical Review of the Use of Spikes in Agile Software Development". The Fourteenth International Conference on Software Engineering Advances, 2019.

[5] D. Leffingwell, Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison- Wesley Professional, 2010.

[6] R. Wirfs-Brock, J. Yoder, E. Guerra, "Patterns to develop and evolve architecture during an agile software project". In Proceedings of the 22nd Conference on Pattern Languages of Programs, 2015.

[7] E. Philippus, "Architecture spikes". Agile Werken ImprovemenT BV, 1, 1-3, 2009. [Online]. Available: http://www.improvement-services.nl/Articles/Architecture%20Spikes.pdf

[8] M. Cohn, *Agile estimating and planning*. Pearson Education, 2005.

[9] E. Miranda, P. Bourque, and A. Abran. "Sizing user stories using paired comparisons." Information and Software Technology, vol. 51, issue 9, pp.1327-1337, 2009.

[10] D. Canty. Agile for project managers. CRC Press, 2015.

[11] A. Mitchell, "A Review of Mixed Methods, Pragmatism and Abduction". In 17th European Conference on Research Methods in Business and Management (p. 269). Academic Conferences and publishing limited, 2018.

[12] M. Saunders, P. Lewis, A. Thornhill, Research Methods for Business Students (7th ed.). Nueva York: Pearson Education, 2016.

[13] P. Boynton, T. Greenhalgh, "Selecting, designing, and developing your questionnaire." Bmj, vol. 328, pp.1312-1315, 2004.

[14] M. Hennink, I. Hutter, A. Bailey, Qualitative research methods. London: SAGE Publication Limited, 2020.

[15] W. Cochran, Sampling techniques. John Wiley, 2007.

[16] S. Mayr, E. Erdfelder, A. Buchner, F. Faul, "A short tutorial of GPower" Tutorials in quantitative methods for psychology, vol. 3, issue 2, pp. 51-59, 2007.

[17] K. Adams, E. Lawrence, Research Methods, Statistics and Application. SAGE Publications, 2018.

[18] J. Yli-Huumo, A. Maglyas, K. Smolander, "How do software development teams manage technical debt?–An empirical study" Journal of Systems and Software, vol. 120, p. 195-218, 2016.

[19] G. Lucassen, F. Dalpiaz, J. van der Werf, S. Brinkkemper, "The use and effectiveness of user stories in practice". In International working conference on requirements engineering: Foundation for software quality, pp. 205-222. Springer, Cham, 2016.

[20] A. Albadarneh, I. Albadarneh, and A. Qusef, "Risk Management in Agile Software Development: A Comparative Study," In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-6. IEEE, 2015.