Research Article

Journal of Electrical Electronics Engineering

A Survey of Software Metrics for Transitioning to an Updated Version of the Current Existing Framework or a New Framework

Md. Rafid Islam[1], Akib Ahmed[2], Ratun Rahman[3]*, Rafsan Jany[4]

[1]Software Engineering Islamic University of Technology (IUT) Gazipur, Dhaka, Bangladesh

[2]Software Engineering, Islamic University of Technology (IUT) Gazipur, Dhaka, Bangladesh

[3]Software Engineering Islamic University of Technology (IUT) Gazipur, Dhaka, Bangladesh

[4]Software Engineering Islamic University of Technology (IUT) Gazipur, Dhaka, Bangladesh

***Corresponding author**
Ratun Rahman, Software Engineering, Islamic University of Technology (IUT) Gazipur, Dhaka, Bangladesh

Submitted: 25 Nov 2022; **Accepted:** 29 Nov 2022; **Published:** 24 Dec 2022

Citation: Islam R, Ahmed A, Rahman R*, Jany R. (2022). A Survey of Software Metrics for Transitioning to an Updated Version of the Current Existing Framework or a New Framework. *J Electrical Electron Eng*, 1(1), 154-158.

Abstract

A software framework is an abstraction that allows to selectively updating software that provides generic functionality by adding additional code written by the user. Basically, a physical or conceptual structure is used to support or guide the construction of something that grows the structure into something productive. Many effective software teams start with frameworks as their foundation. Bundled code libraries, software modeling, APIs, and a myriad of other capabilities, which are included in these frameworks, make programming considerably faster and more effective. Therefore, software engineers often delve into various frameworks for development and other purposes. A survey has been conducted and opinions from teachers, students, and developers were solicited. This paper examines the results of the poll and tries to explain what software engineers, students, or seasoned developers experience when switching to a new framework in the middle of a project.

Keywords: Software Metrics, Software Frameworks, Domain, Software Tools, Update Version, Headache

Introduction

Throughout our internship, we were exposed to a variety of frameworks that we were unfamiliar with we quickly found that transitioning to new technology is not easy. We were apprehensive about doing so. It was also inconvenient because we were already familiar with a framework that we had employed in our academics. For example, we learned React but saw Angular in action and both are JavaScript frameworks but still, we found it a bit of a challenge.

Learning a framework and becoming familiar with it is crucial for software developers as it can be advantageous in various ways. For example, each of the JS web framework solutions has its own set of benefits and robust capabilities that assist in the production of efficient, easy-to-understand code. They also pave the way to program your website with less time spent on development, debugging, and testing. However, moving on to a new framework or software tool when a developer has grown accustomed to one and must move to another is a barrier. It can have an impact on their overall performance.

An effort has been made in this paper to take on the task of analyzing a survey that includes data, pie charts, and graphs that show the opinions of the people who have experienced a shift in framework or software tools while working on a project. Some of the questions

Covered in the poll included whether the developers were comfortable switching frameworks or not, if they had received any aid from colleagues, if they achieved their goal, did it improve their skills, and whether they received a salary boost or not. We will look at the results of the survey to evaluate if investing the effort into moving to a new framework was worthwhile after all.

Research Methodology

The methodology approach of this research is based on the observational method, capturing data through observation of a behavior or activity obtained in real- time. Here we captured the data based on human surveillance and their responses through a survey. This modeling has been chosen for this research because it provides a systematic method for description, exploration, and analysis of the dynamic behavior of the complex system. Observational is a very popular system for many researchers for studying and monitoring a problem. The concept of a feedback system reveals the majority decision of people and predicts what other people would most likely feel. We have used Google form to record our data because it contains different types of diagrammatic tools. These tools can help us visualize the data graphically and more comfortably.

Data Collection

The model developed for this work is based on data mainly collected from IUT Software Engineering final year students and their respective internship companies in Bangladesh. The questions and relevant choices have been set according to our topic 'A headache for software engineers as they transition to an updated version of the current existing framework or a new framework.' These questions were created by six of our software engineering students from the Islamic University of Technology (IUT) that focused on the difficulties, struggles, and outcomes of learning a new framework that a software engineer faces.

Using these factors mentioned above, we have developed and distributed a set of questionnaires to all the people connected to software engineering we know. We have managed to get 143 responses. The criteria for selection of this preliminary study was people who are related to software engineering with at least four months experience.

Data were mainly collected throughout two months in 2021 (September-October). To ensure the quality of data, team members focused on gathering data from their internship companies and closed people only associated with software engineering.

We have centered our survey based on legitimate responses in a smaller number. However, more people are willing to participate in this survey whom we may have missed and the plan is to take as much valid feedback as possible.

Questionnaire

Our questionnaire is divided into five sections. We used this series of questions to try to figure out what our survey's purpose was. In a nutshell, we will explain why we asked these questions below.

Section 1

In this section, we tried to get some basic information from them, such as their name, current employment, and experience in their industries. We inquired about their earlier experience to learn how they dealt with challenges and what advantages/disadvantages they had as a result of it.

Section 2 to 4

We constructed a comparable set of questions in sections 2, 3, and 4 based on three primary subjects of our survey: domain shift, learning new software tools, and updating to a new version. To begin, a change in domain occurs when a person switches from one language or framework to another. When a Python programmer needs to switch to JavaScript, for example. Second, learning a new software tool (IDE) necessitates changing the platform on which one has been working. When a person who has worked with Blender must switch to Adobe Illustrator, for example. Finally, update to a new version means when an existing project needs to be updated with the latest version. For example, a project that was initially developed with Angular 1 needs to be updated to Angular version 2. On a scale of 1 to 5, we wanted to know if the person had to change those three categories, how comfortable they were doing so, and how helpful their coworkers were, with 1 being the least likely and 5 being the most probable. We also gathered information on the estimated time spent learning, the key learning source, and the learning goal.

This section summarizes the learning process in general. We gathered the participants' opinions on the importance of learning for their jobs. We also wanted to see whether their worth had increased and if we might get a better deal. Finally, we wanted to determine if the participants were willing to learn and if the learning was worthwhile for them/

Data Analysis

We have selected the key factors based on the priority of relevancy and usability for the approach in this research. Initially, there were more than 35 factors for preliminary analysis, as we have taken every possible combination of data we can collect. However, the factors have been reduced to 26 after primary research and based on effectively and efficiency. After hearing multiple feedbacks from initial reports on questionnaires, we were able to cut down some of them for our goal.

The procedure followed to extract people’s perception of the productivity influence factors:

- Convert the qualitative scale to quantitative one; five divisions of scale from one to five where 1 is low likely or not effective and 5 is highly likely or most effective.
- We have divided our survey into three major parts: changing domain, learning new software tools, and updating to a new version.
- Through various questions, we tried to figure out the difficulties, struggles, and the effectiveness of the process consisting of yes, no, and quantitative questions.

Survey Results

After conducting the survey, we are representing the data with tables, graphs, and pie charts.

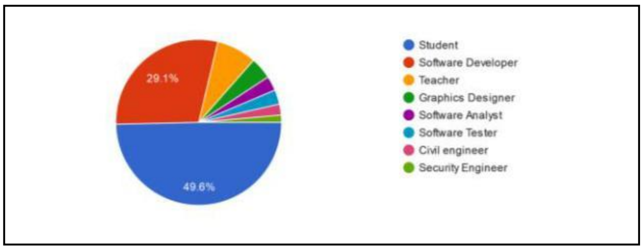

Figure 1: Most of the respondents are students. Software developers are the next major participants. Most of them are people whom we have met while working as interns in various companies.

TABLE I. TANSITIONING WITH EXPERIENCE

	Changing Domain		Learn New Software Tools		Update new Version	
Exper ience	Comfor tability	Coope ration	Comfor tability	Coope ration	Comfor tability	Coope ration
<1 Year	76.1%	56.6%	78.3%	73.7%	65.2%	64.5%
1-5 years	71.2%	71.5%	57.2%	72.1%	74.4%	63.9%
5-10 years	68.3%	69.9%	54.4%	71.2%	72.6%	62%
0+ years	61.2%	69.3%	51.8%	70.9%	72.4%	62.1%

Table 1: A representation of peoples' comfortability and cooperation with their growing experience in three different factors-changing domain, learning new software tools, and updating new versions. After analyzing the data we realize that inexperienced developers face the most difficulties and with a growing experience, they become a bit more comfortable. However, after a certain period, people become less comfortable in adapting to a new system.

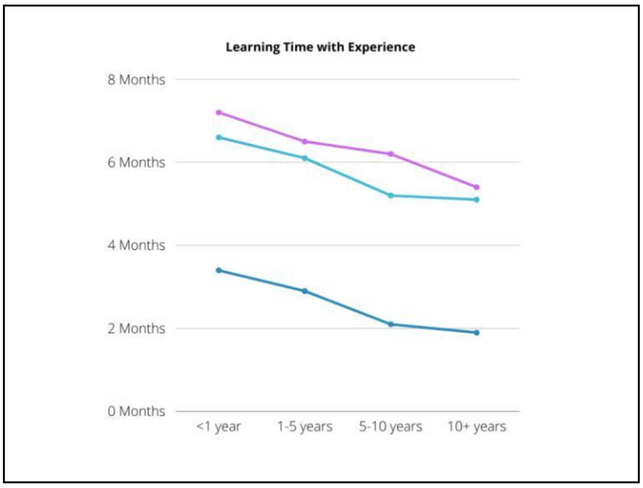

Figure 2: The graph depicts the progression of learning time as a function of experience. The more experienced a developer is the less time he takes to learn a software tool or framework.

Figure 3, 4 & 5: From these pie charts, we can see that quite a lot of people had to change their domain, or learn a new software tool or an updated version of a framework while working on a project out of necessity. Most of the respondents have learned a new software tool, followed by a change of domain and then learning an updated version of the framework

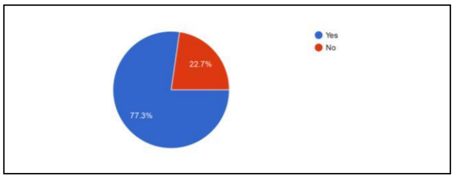

Figure 3: Answer of 'Have you changed your domain while working?'

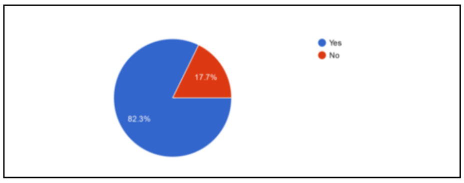

Figure 4: Answer of 'Have you learned a new tool while working?'

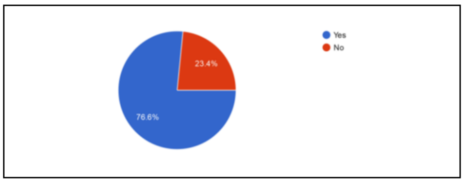

Figure 5: Answer of 'Have you changed towards a completely new version while working?'

Figure 6 shows that the most preferred learning source is YouTube, followed by Stack Overflow and people are reluctant to open a book for fast learning.

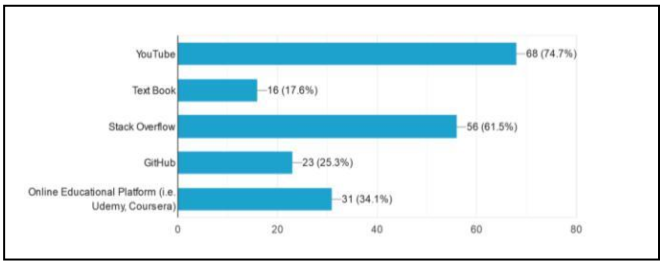

Figure 6: Answer of 'Which platform have you used to learn the new module?'

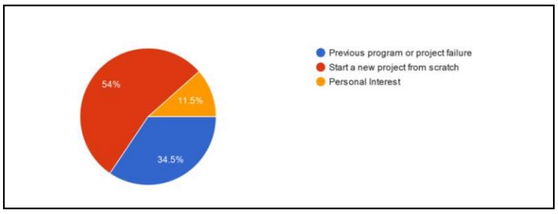

Figure 7: The chart shows that the most preferred learning source is YouTube, followed by Stack Overflow and people are reluctant to open a book for fast learning.

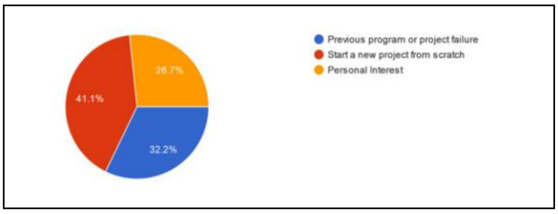

Figure8: Reason for learning newsoftware

Figure 9, 10 & 11: Purposes of learning these processes are graphically represented in these pie charts. In the case of the domain, starting a new project from scratch was the main purpose (54%), then previous program failure (34.5%), and then personal interest (11.5%). For learning a new software tool, starting a new project from scratch was also the main purpose (41.1%) followed by previous project failure (32.2) and personal interest (26.7%). However, in the updated version previous program failure is the main reason (63.1%) followed by starting a new project from scratch (23.8%) and personal interest (13.1%).

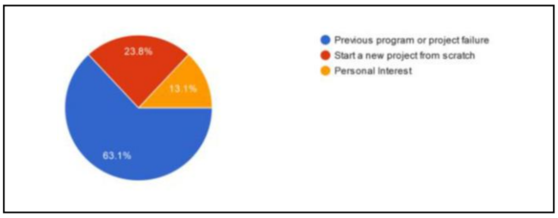

Figure9: Reason for version change

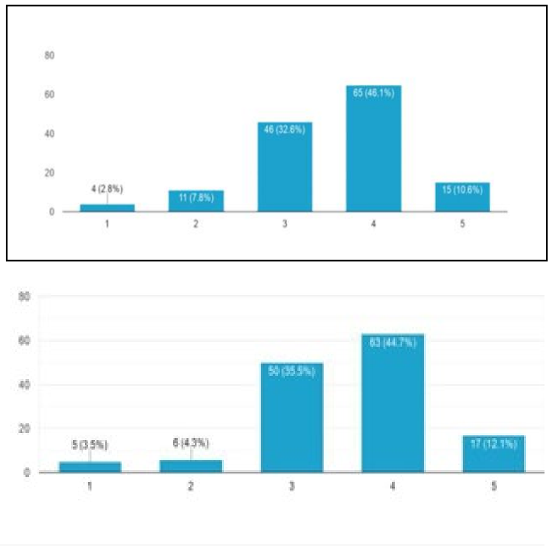

Figure 10, 11: People’s value after Transitioning

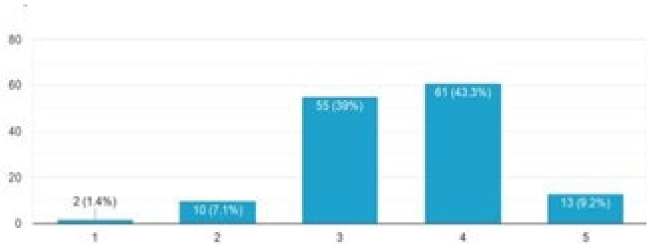

Figure 12: It depicts people's perceptions of how useful and relevant the learning experience was for their work. According to the chart, most people believe that learning a new framework is helpful (46.1%).

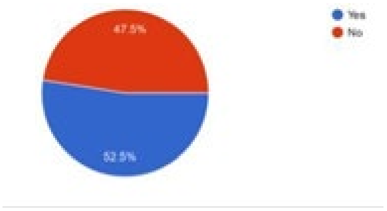

Figure 13: It shows how people see their own worth after learning a new framework. According to the chart, most people believe that their value has increased after the process. We can draw a conclusion that learning or shifting towards a new framework was worthwhile.

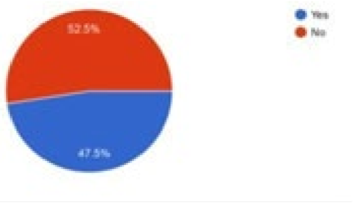

Figure 14: according to the pie chart, we can say most people did get a better offer after learning a new framework.

Related Works

This study is part of a category of online-based surveys but the specific point of view is different. To place the research presented here in context, we first give an overview of the study, the purpose of the survey as well as the survey methodology then analyze the survey results.

There has been some work that can be indirectly related to the study. For example, reported a survey of Software Metric used in Research Software Development. Perform a case study to identify the main factors that influence productivity and how they affects agile teamwork through interviews, surveys, and literature reviews [1-3].

Albert presents a case study that is a comparison between companies adopting Agile Methods (mainly XP) with companies using document-driven approaches from a well-defined point of view: requirements management and uncertainty management [4].

Evaluated the ability of the Software source code analysis process and tools to predict possible defects, errors, or problems in the software products. In their study, focuses on the importance of software metrics and discusses their key role in software development processes perspective to Sri Lankan’s organizations, and systematically analyses the importance of software metrics in software development projects [5-7].

Figure 15: We can observe that the majority of people were hesitant to adopt a new framework.

Limitations of the Study

There are many limitations to this survey. Firstly, the survey was conducted for a few days. In the meantime, it was hard to approach for feedback as the global

Pandemic is going on right now. Furthermore, we got 143 responses. The main source of data came from our university bodies as well as the company we had done our internships with As a result; we got various people from different fields responding to our survey.

Another limitation is the lack of response from experts. It is hard to draw a picture for the whole software industry from our survey because the responses we got from mostly inexperienced people. They might not represent the overall picture of our topic.

Finally, the survey topic might have been confusing for some people. They might be giving their opinion without properly grasping the whole scenario.

Conclusion and Future Works

By analyzing the survey result, it can be quite hard to draw any kind of conclusion for the whole software industry. As there are many inexperienced responses included, we can say that it helps to see an overall image from the future of software developers and others of the related fields. It is also seen that the qualified personnel tend to stick with their ability to work at their current position. On the other hand, the inexperienced ones are eager to learn more and take on new challenges. They can get support from the experts and express themselves. Whether it is learning a new framework or new tools, we can see that it is helpful for their career growth. In the modern era, people in the whole world are getting advanced in technology and there is almost no alternative [8-14].

Some large surveys could be conducted as well as recommendations for overcoming the problem could be made in the future. There are a lot of scopes for help regarding this topic. Nowadays there are a lot of frameworks for software developers, graphics designers, and many others. People often find it hard to choose the best option for them. Therefore, it is good that they can know about the pros and cons of changing to a new framework for their job. These will help the recruiters as well as the applicants in the future.

References

1. Riguzzi, F. (1996). A survey of software metrics. Università degli Studi di Bologna.
2. Grossman, T., Fitzmaurice, G., & Attar, R. (2009, April). A survey of software learnability: metrics, methodologies and guidelines. In Proceedings of the sigchi conference on human factors in computing systems (pp. 649-658).
3. Agnihotri, M., & Chug, A. (2020). A systematic literature survey of software metrics, code smells and refactoring techniques. Journal of Information Processing Systems, 16(4), 915-934.
4. Timóteo, A. L., Álvaro, A., De Almeida, E. S., & de Lemos Meira, S. R. (2008). Software metrics: A survey. Sl: sn.
5. Padmini, K. J., Bandara, H. D., & Perera, I. (2015, April). Use of software metrics in agile software development process. In 2015 Moratuwa Engineering Research Conference (MERCon) (pp. 312-317). IEEE.
6. Filó, T. G., Bigonha, M., & Ferreira, K. (2015). A catalogue of thresholds for object-oriented software metrics. Proc. of the 1st SOFTENG, 48-55.
7. Grossman, T., Fitzmaurice, G., & Attar, R. (2009, April). A survey of software learnability: metrics, methodologies and guidelines. In Proceedings of the sigchi conference on human factors in computing systems (pp. 649-658).
8. Mens, T., & Tourwé, T. (2004). A survey of software refactoring. IEEE Transactions on software engineering, 30(2), 126-139.
9. Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. Working group reports on ITiCSE on Innovation and technology in computer science education, 204-223.
10. Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005, September). Managing uncertainty in requirements: a survey in documentation-driven and agile companies. In 11th IEEE international software metrics symposium (METRICS'05) (pp. 10-pp). IEEE.
11. Zoubi, Q., Alsmadi, I., & Abul-Huda, B. (2012, May). Study the impact of improving source code on software metrics. In 2012 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 1-5). IEEE.
12. Fernando, W.K.S.D., Wijayarathne, D.G.S.M., Fernando, J.S.D., Mendis, M.P.L. and Guruge, I., (2014). The importance of software metrics: Perspective of a software development projects in Sri Lanka. In SAITM Research Symposium on Engineering Advancements (pp. 91-95).
13. Libguides., Macalester Edu. (2021). All Guides Data Module. What is Research Data?.
14. Eisty, N. U., Thiruvathukal, G. K., & Carver, J. C. (2018, October). A survey of software metric use in research software development. In 2018 IEEE 14th international conference on e-Science (e-Science) (pp. 212-222). IEEE.