Size Measurement and Effort Estimation in Microservice-based Projects: Results from Pakistan

Görkem Kılınç Soylu[1], Hüseyin Ünlü[1], Isra Shafique Ahmad[1] and Onur Demirörs[1]

[1] Izmir Institute of Technology, Gülbahçe, Urla, Izmir, 35430, Türkiye

Abstract

During the last decade, microservice-based software architecture has been a common design paradigm in the industry and has been successfully utilized by organizations. Microservice-based software architecture, specifically in the form of reactive systems, has substantial differences from the more conventional design paradigms, such as the object-oriented paradigm. The architecture moved away from being data-driven and evolved into a behavior-oriented structure. The usage of a single database is replaced by the structures in which each microservice is developed independently and has its own database. Therefore, adaptation demands software organizations to transform their culture. In this study, we aimed to get an insight into how Pakistani software organizations perform size measurement and effort estimation in their software projects which embrace the microservice-based software architecture paradigm. For this purpose, we surveyed 49 Pakistani participants from different agile organizations over different roles and domains to collect information on their experience in microservice-based projects. Our results reveal that although Pakistani organizations face challenges, they continue using familiar subjective size measurement and effort estimation approaches that they have used for traditional architectures.

Keywords

Microservices, software size measurement, effort estimation, agile, survey, Pakistan

1. Introduction

During the last decade, we are experiencing a paradigm shift in the software industry; a growing number of software organizations have preferred MicroService-based Software Architectures (MSSA) in their projects. Today, many massive software organizations such as Amazon, LinkedIn, Netflix, SoundCloud, and Uber have adopted the MSSA paradigm [1].

MSSA consists of loosely coupled distributed microservices that work in cohesion. These microservices communicate by messages over lightweight mechanisms, such as REST [2]. Each microservice can be developed, deployed, and scaled independently [1], [3], [4]. The change in one microservice should not affect another microservice. MSSA has also been associated with event-based asynchronous service communication and ledger-style data persistence to reveal real-world benefits regarding scalability, reliability, and performance [5].

The essential characteristics of microservices are found to fit into the agile development methodology [6]–[8]. Delivering working software frequently, which is one of the most important principles of the Agile Manifesto, can be provided by independent microservices with a single business capability. Additionally, an agile team is expected to find it easier to grasp domain concepts within a bounded context, as they are interrelated [8].

MSSA exhibits significant distinctions from the conventional object-oriented paradigm, particularly in the context of reactive systems [9]. We observed that traditional modeling

IWSM Mensura 2023, September 14–15, 2023, Roma, Italy

gorkemkilinc@iyte.edu.tr (G. Kılınç Soylu); huseyinunlu@iyte.edu.tr (H. Ünlü); israshafique@iyte.edu.tr (I. Shafique Ahmad); onurdemirors@iyte.edu.tr (O. Demirörs)

0000-0002-7047-0556(G. Kılınç Soylu); 0000-0001-9906-6066 (H. Ünlü); 0000-0001-6601-3937 (O. Demirörs)

notations used for analysis and design are ineffective when working with microservices [5]. Even the fundamental abstraction in use today, the 'object' in the Object-Oriented methodology, needs to change. Consequently, the new paradigm requires a shift similar to the one experienced during the transition from structured and object-oriented analysis and design [9]. This new paradigm necessitates adapting to the service as the primary abstraction, event-based decomposition, and asynchronous communication, which are fundamental characteristics of reactive microservices.

Similarly, we observed that traditional functional size measurement methods may not align well with the structure of the microservices [10], [11]. MSSA moved away from being data-driven and evolved into a behavior-oriented structure. The usage of a single database is replaced by the structures in which each microservice is developed independently and has its own database. It is important to make software size measurement methods compatible with the architecture to be measured. However, lightweight requirements are the only artifacts for early-phase estimations in the agile world. Therefore, a size measurement method compatible with MSSA should also consider the lack of detailed requirements in the agile world.

We previously performed a survey [12] and an interview [13] to explore the organizational choices and the challenges of the agile development process, agile tools, software analysis, design, test, size measurement, and effort estimation. We presented the views, experiences, and challenges of participants over different roles, domains, and countries. The results showed that organizations globally continue to use the same analysis, design, size measurement, and effort estimation approaches that they were using previously in traditional architectures. However, they face unique challenges in the management of MSSA projects.

As a developing country, Pakistan's market for computer software has seen steady growth for the past several years. Much of the growth is driven by the work of freelancers who earned subpar certifications from local institutions and support specialized fields in the local market. Pakistan is currently the fourth largest freelance provider after India, Bangladesh, and the United States, with gaming software development [14].

In this study, we aimed to explore how Pakistani Agile organizations perform size measurement and effort estimation in their projects developed with MSSA. For this purpose, we performed our previous survey of over 49 Pakistani participants from different organizations, roles, and domains. We compared the results of Pakistani organizations with our previous studies performed with participants from different countries. We observed similar results that are employed globally.

2. Related Work

In the literature, there are a number of studies analyzing migration issues or technical issues for converting monoliths to microservice-based systems. However, only a few studies explored the issues from the project management perspective and for building new microservice-based systems (please see [13] – Section 2, due to page limitations). We could not find enough studies to perform a systematic literature review on the analysis, design, size measurement, and effort estimation of microservices. To the best of our knowledge, there is no study that reveals Pakistani organizational practices in terms of size measurement and effort estimation for microservice-based projects.

3. Research Method

In this study, we conducted a survey to get an insight into how Pakistani software organizations perform size measurement and effort estimation in their microservice-based software projects. On the basis of the answers we collect from the practitioners, we aim to identify the trends in the methods and technologies used in conducting microservice-based projects. The research questions we address in this study are as follows:

RQ1. How do organizations in Pakistan apply the agile methodology in microservice-based projects?

RQ2. How do organizations in Pakistan perform size measurement and effort estimation in microservice-based projects?

RQ3. Are the size measurement and effort estimation methods and technologies utilized in Pakistan different from those employed globally? If there are differences, what sets them apart?

Due to the page limitations, we provided the research method section in our online supplementary material.

4. Results

In this section, due to space constraints, we report a summary of the survey results of 49 participants regarding the demographics, experience with Agile methodology and microservice-based projects, and size measurement and effort estimation.

4.1. Demographics of the participants

The majority of our participants, i.e., 60%, completed their undergraduate degree in computer science, while 30% graduated from software engineering, 4% from information systems, and % six from other fields. 68% of our participants completed their graduate degree, i.e., 38% in computer science, 20% in software engineering, and 10% in other fields.

The participants in our study hold diverse positions in their respective companies, including developer, senior developer, software architect, test engineer, software engineer, and other roles such as software test engineer, analyst, and chief executive officer, among others. Most of the participants (38%) work as a senior developer, while 26% work as a developer, 6% as a software engineer, 6% as a software architect, 6% as a software test engineer, and the remaining 18% work in other roles.

The survey participants come from organizations across various industry sectors. We grouped some very specific answers under more general domains to achieve more informative and readable charts and consequently to enable meaningful deductions from the results. According to the answers, 34% of our participants work in the finance sector, while 24% work in a field related to mobile software, 16% work in web development, 14% work in the entertainment sector, 6% work in game development, and 6% work in telecom.

The participants' experience in their current role ranges from 1 to 10 years. The mean value of the distribution is 2.74, whereas the median is 2.5. Their experience in software engineering-related fields also ranges from 1 to 10 years. The mean value of the distribution is 4.29, whereas the median is 4. The detailed charts and distributions can be seen in the online supplementary material.

4.2. Experience with Agile methodology and microservice-based projects

Our results show that the participants' organizations' experience with agile methodologies ranges from 1 to 23 years, whereas their experience in microservices ranges from 1 to 9 years. Table 3 shows the distribution. The mean and the median of the organizations' experience in agile methodologies are 7.81 and 5, whereas the mean and the median of the organizations' experience in microservices are 4.03 and 3.

In our survey, we asked our participants which agile methodology they use in their organizations. The results show that the majority of the participants use SCRUM, and the only other methodology used by the participants is Kanban. No other methods were mentioned by any of our participants, while some indicated that they use SCRUM and Kanban together. According to our results, 80% of the participants use SCRUM, and 25% use Kanban. Notice that the intersection is not an empty set.

Our survey included some questions about the work environment and team dynamics. The results show that the team members of most of the participants work on many projects simultaneously. 58% of the participants work on multiple projects at the same time, whereas 42% work on only one project. The majority of our participants (84%) indicated that everyone is easily accessible while communicating within the team and with upper management. Only 16% indicated that there is a hierarchy in communication. According to the answers of the participants, the majority of them, i.e., 84%, have team members who are certified in agile methodologies (such as being a SCRUM Master). The detailed charts can be seen in the online supplementary material.

4.3. Size measurement and effort estimation

In this section, we will give a summary of the results concerning the size measurement and effort estimation that we gathered via our survey.

We asked our participants how they perform effort estimation in their microservice-based projects. Half of the participants indicated that they use expert judgment. 30% of the participants stated that they do effort estimation ad-hoc, while 28% use an organization-wide prediction model, 24% use planning poker, 18% use parametric estimation method, and 4% use wide-band delphi. It was possible to give multiple answers to this question. It is seen from the results that some participants use multiple methods, whereas some do not perform any effort estimation (see Fig. 1a).

The next question of the survey concerns the software size measurement method used in microservice-based projects. The results show that most participants (58%) use the story points method. Followingly, 28% utilize case points, while 14% of the participants use source lines of code, 6% use t-shirt size, 2% use COSMIC, 4% use other methods, and 12% don't perform any size measurement (see Fig. 1b).

We also investigated in our survey which tasks are included in effort estimation. According to the answers, development effort is considered in effort estimation by 88% of the participants' organizations. Test effort is included in effort estimation by 70%, while design effort is included by 60%, analysis effort is included by 56%, and operations effort is included by 48%. Only 4% of participants indicated that they do not perform any effort estimation (see Fig. 1c).

As for the smallest unit that the organizations estimate/record effort, the results show that the majority of the organizations (62%) estimate/record requirements. 48% estimate/record product backlog items, while 18% perform Epic-based effort estimation (see Fig. 1d).

Our survey shows that the majority of our participants do not use any tool for effort estimation or size measurement, whereas only 38% use a tool for this purpose (see Fig. 1e).

Our survey includes a question about the challenges observed during the effort estimation process when utilizing microservice-based projects. For this question, participants were expected to select a number between 1 and 5, 1 being the least and 5 being the most. 6% of the participants do not observe any challenges, whereas 10% observe only some challenges. The majority of the participants (54%) indicated that they observe a medium level of challenges, whereas 24% find effort estimation very challenging, and 6% find it extremely challenging (see Fig. 1f).

Our participants were also asked about the change in the precision of estimated effort when utilizing microservice-based projects. Again, they were expected to choose a number between 1 and 5, 1 being the least and 5 being the most. 6% indicated no change in precision, whereas 10% indicated some changes. Half of the participants chose number 3, meaning that they observed a medium level of change in the precision of estimated effort when utilizing microservice-based projects. 30% indicated a high level of change, whereas 4% indicated a very high level of change (see Fig. 1g).

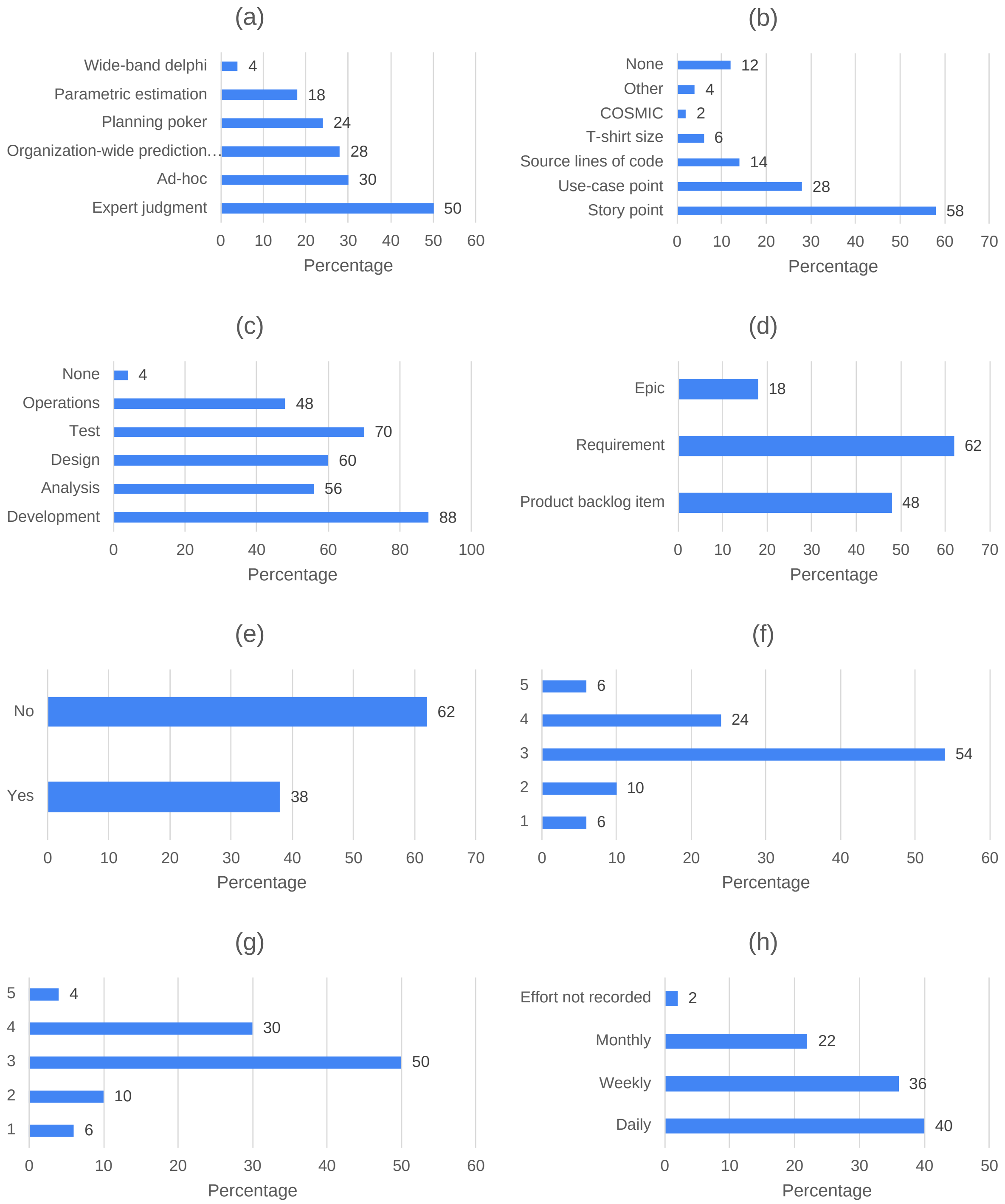

Figure 1: The percentage of the responses regarding (a*) the effort estimation method, (b*) the size measurement method, (c*) included tasks in effort estimation, (d*) the smallest unit to record/estimate effort, (e) whether a tool used for effort estimation, (f) the challenge level of the effort estimation, (g) the change level in the precision of effort estimation, and (h) the record frequency of actual individual effort (*: multiple answers)

Lastly, we asked in our survey how frequently the actual individual effort is recorded in their organizations. 40% answered that they record daily, whereas 36% record weekly and 22% record monthly. Only 2% indicated that they do not record the actual individual effort (see Fig. 1h).

5. Discussion

In this study, we explored how Pakistani software organizations perform size measurement and effort estimation in their software projects which embrace the MSSA paradigm in the agile world. The survey results show that there is no de facto approach utilized for these tasks among the participants.

In general, organizations utilize subjective size measurement (such as story point) and effort estimation (such as expert judgment) methodologies for microservice-based projects. However, these subjective methodologies are also heavily criticized [17] due to the misinterpreted perceptions of size, complexity, subjectivity, and team-specific characteristics, which make them hard to transfer across teams. Therefore, effort estimation in Agile organizations becomes a challenging task.

Functional size measurement methods have been widely used for objective size measurement of monolith software architectures. To illustrate, COSMIC, one of the functional size measurement methods, is based on counting data movements [18]. The method suggests using data analysis approaches such as relational data analysis, class diagram, and entity relationship analysis. However, the use of traditional data analysis methods has gradually decreased in microservice-based projects where requests and responses are the majority instead of data flow. The architecture moved away from being data-driven and evolved into a behavior-oriented structure. The usage of a single database is replaced by the structures in which each microservice is developed independently and has its own database. It can be argued that functional size measurement methods, such as COSMIC, may be suited for MSSA by counting the messages between microservices as data groups. However, MSSA, in the reactive form, is not the same as Service Oriented Architecture (SOA). Microservices are designed using event-based architecture. Events, by being used in asynchronous communication of different services, led to faster decision-making, reduced costs, and improved scalability in microservices. This asynchronous communication is generally managed by structures such as event queues. Microservices subscribe to these event queues to consume the published events by other microservices. Therefore, there is a need for a method to measure the size of microservice-based projects, using events as base measurement units which are fundamental concepts of the MSSA. We previously proposed an event-based size measurement method for MSSA that counts the number of process, communication, and interaction events. We built more precise effort estimation models using the event-based size compared to the models built with COSMIC and source lines of code [10], [11].

The main problem in today's agile software industry is that organizations do not want to spend time and resources on size measurement and effort estimation. It is a fact that functional size measurement methods need expertise; therefore, especially small-sized organizations cannot create precise effort estimation models using these size measures. On the other hand, there is an opportunity to predict the functional size of the project requirements using artificial intelligence. In [19], we attempted to predict the COSMIC size of use case descriptions using Natural Language Processing (NLP) models. Although we used a relatively small dataset to train the models, we obtained promising size prediction results with a 0.23 Mean Magnitude of Relative Error (MMRE) and 0.75 PRED(30). Agile organizations can benefit from this prediction model to perform an early estimation.

Another main problem is that agile organizations generally do not have historical effort data to construct effort estimation models. The size of the software project can be measured at every stage of the project, from the requirements, code, or user interface. However, accessing historical effort data from previous projects may not always be possible within organizations. Therefore, it can be challenging to estimate effort for new projects. In such cases, software organizations may use software benchmarking datasets to construct effort estimation models or organizational benchmarking purposes [20], [21].

In this study, we also aimed to answer whether the size measurement and effort estimation methods and technologies utilized in Pakistan are different from those employed globally (see Table 4). The survey results show that Pakistani organizations also apply subjective size measurement and effort estimation methods, as employed globally. They mainly use the globally employed size measurement method, story point, in their microservice-based projects. However, the major effort estimation method in Pakistan is different from the global results. Expert judgment is the mostly used effort estimation methodology in Pakistani organizations while planning poker is the common global approach. In our previous interview [13], participants generally stated that teams first play Planning Poker to estimate the effort. However, if the estimated effort is found as not reasonable by the upper management, the estimation is updated by the managers. This scenario may also be applicable to Pakistani organizations.

Table 4
Comparison of the results

Question	Pakistan	Global [12], [13]
Effort Estimation Method	Expert Judgment (50%)	Planning Poker (31%)
Size Measurement Method	Story Point (58%)	Story Point (38%)
Most Included Task in Effort Estimation	Development (88%)	Development (91%)
Smallest Unit to Estimate/Record Effort	Requirement (62%)	Product Backlog Item (40%)
Tool for Effort Estimation	No (62%)	No (50%)
Challenge Level of Effort Estimation	3 (54%)	3 (35%)
Change Level in the Precision of Effort Estimation	3 (50%)	3 (51%)
Actual Effort – Record Frequency	Daily (40%)	Not Recorded (35%)

Also, the majority of Pakistani organizations do not see any challenge or precision difference between the effort estimation of microservice-based projects and traditional monolith projects. This may be related to employing professionals who analyze team dynamics well and are experienced in microservice-based projects. Or, team members may not fully reflect their actual efforts due to competition concerns, as indicated in [13].

6. Limitations and threats to validity

Our study is not without limitations. The number of participants in the survey may be considered a limitation. We have done our best to increase our sample size while making sure that a single organization will not dominate the survey. This is one of the factors that limit the number of participants; we have also paid attention to making sure participants were from a wide range considering their experience, role, and organization size. As a result, we ended up with a good representation but a limited number of participants. Although it is difficult to generalize the results as best practices, we believe we provide a meaningful snapshot of Pakistan's software industry that utilizes microservices.

A possible potential threat can be related to misconceptions about questions, as there is no face-to-face communication during the survey. We performed a pilot study with 3 participants to prevent this potential threat. We eliminated misunderstandings and ambiguities and ensure the interview was well-designed to serve our goal.

Another validity threat can be related to whether correct conclusions are reached through rigorous and repeatable treatment. We conducted our survey with participants from different roles, domains, and experiences without losing control of the participants to prevent the participation of unrelated participants. We also set criteria to assess the quality, accuracy, and trustworthiness of the survey results. We eliminated some results due to a lack of experience in MSSA.

7. Conclusion

During the last decade, MSSA has been a common design paradigm in the industry and has been successfully utilized by organizations. MSSA, specifically in the form of reactive systems, has substantial differences from the more conventional design paradigms, such as the object-oriented paradigm. The architecture moved away from being data-driven and evolved into a behavior-oriented structure. The usage of a single database is replaced by the structures in which each microservice is developed independently and has its own database. Therefore, adaptation demands software organizations to transform their culture.

In this study, we aimed to explore how Pakistani Agile organizations perform size measurement and effort estimation in their projects developed with MSSA. For this purpose, we performed our previous survey of over 49 Pakistani participants from different organizations, roles, and domains. We compared the results of Pakistani organizations with our previous studies performed with participants from different countries [12], [13]. The results show that Pakistani organizations also apply subjective size measurement and effort estimation methods in their microservice-based projects, as employed globally.

This survey provided insight for Pakistani software organizations to transform their culture for microservice-based software architectures. In the meantime, we have observed that we were not able to observe many of the insights gained by the practitioners due to the nature of questionnaire-based surveys for domains that experience transformation, although we provided a free text field for the “Other” choice in the answers and a field to enter the comments. We have seen that many participants did not enter any further comments during the survey. Therefore, a follow-up interview can be performed with Pakistani organizations as future work.

8. Supplementary Material

The supplementary material, including the removed sections, charts, and survey questions, can be found at *https://bit.ly/46Zz2PK*.

9. Acknowledgments

This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 1001 [Project number: 121E389] program.

10. References

[1] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, ‘Microservices’, IEEE Software, vol. 35, no. 3, Art. no. 3, May 2018, doi: 10.1109/MS.2018.2141030.

[2] A. Ivanchikj, C. Pautasso, and S. Schreier, ‘Visual modeling of RESTful conversations with RESTalk’, Softw Syst Model, vol. 17, no. 3, Art. no. 3, Jul. 2018, doi: 10.1007/s10270-016-0532-2.

[3] J. Thönes, ‘Microservices’, IEEE Software, vol. 32, no. 1, Art. no. 1, Jan. 2015, doi: 10.1109/MS.2015.11.

[4] A. R. Sampaio et al., ‘Supporting Microservice Evolution’, in 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), Sep. 2017, pp. 539–543. doi: 10.1109/ICSME.2017.63.

[5] H. Unlu, S. Tenekeci, A. Yıldız, and O. Demirors, ‘Event Oriented vs Object Oriented Analysis for Microservice Architecture: An Exploratory Case Study’, in 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Sep. 2021, pp. 244–251. doi: 10.1109/SEAA53835.2021.00038.

[6] L. Barroca, H. Sharp, D. Salah, K. Taylor, and P. Gregory, ‘Bridging the gap between research and agile practice: an evolutionary model’, Int J Syst Assur Eng Manag, vol. 9, no. 2, pp. 323–334, Apr. 2018, doi: 10.1007/s13198-015-0355-5.

[7] V. Garousi, A. Coşkunçay, A. Betin-Can, and O. Demirörs, ‘A survey of software engineering practices in Turkey’, Journal of Systems and Software, vol. 108, pp. 148–177, Oct. 2015, doi: 10.1016/j.jss.2015.06.036.

[8] S. Newman, Building microservices. O’Reilly Media, Inc., 2021.

[9] J. Bonér, Reactive Microsystems. O’Reilly Media, Inc., 2017.

[10] H. Ünlü, T. Hacaloglu, F. Büber, K. Berrak, O. Leblebici, and O. Demirörs, ‘Utilization of Three Software Size Measures for Effort Estimation in Agile World: A Case Study’, in 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2022, pp. 239–246. doi: 10.1109/SEAA56994.2022.00045.

[11] H. Ünlü, T. Hacaloglu, N. Küçükateş Ömüral, N. Çalışkanel, O. Leblebici, and O. Demirors, ‘An Exploratory Case Study on Effort Estimation in Microservices’, in 2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Sep. 2023.

[12] H. Ünlü, B. Bilgin, and O. Demirors, ‘A survey on organizational choices for microservice-based software architectures’, Turkish Journal of Electrical Engineering and Computer Sciences, vol. 30, no. 4, pp. 1187–1203, Jan. 2022, doi: 10.55730/1300-0632.3843.

[13] H. Ünlü, D. E. Kennouche, G. Kilinc Soylu, and O. Demirörs, ‘Microservice-Based Projects in Agile World: A Structured Interview’, Manuscript submitted to Information and Software Technology.

[14] ‘Pakistan - Computer Software’, Official Website of the International Trade Administration, Oct. 11, 2022. https://www.trade.gov/country-commercial-guides/pakistan-computer-software (accessed Jul. 22, 2023).

[15] F. Shull, J. Singer, and D. I. Sjøberg, Guide to advanced empirical software engineering. Springer, 2007.

[16] J. Linaker, S. M. Sulaman, M. Höst, and R. M. de Mello, ‘Guidelines for conducting surveys in software engineering v. 1.1’, Lund University, 2015.

[17] T. Hacaloğlu and O. Demirörs, ‘Challenges of using software size in agile software development: A systematic literature review’, Academic Papers at IWSM Mensura 2018, 2018.

[18] ‘COSMIC Measurement Manual Version 5.0’, The Common Software Measurement International Consortium, 2021. [Online]. Available: https://cosmic-sizing.org/publications/measurement-manual-v4-0-2/

[19] S. Tenekeci et al., ‘Predicting Software Functional Size Using Natural Language Processing: An Exploratory Case Study’. Available: https://www.researchgate.net/publication/372524319_Predicting_Software_Functional_Size_Using_Natural_Language_Processing_An_Exploratory_Case_Study

[20] H. Ünlü et al., ‘Software Effort Estimation Using ISBSG Dataset: Multiple Case Studies’, in 2021 15th Turkish National Software Engineering Symposium (UYMS), Nov. 2021, pp. 1–6. doi: 10.1109/UYMS54260.2021.9659655.

[21] O. R. Yürüm, H. Ünlü, and O. Demirörs, ‘Towards the Construction of a Software Benchmarking Dataset via Systematic Literature Review. Available: https://www.researchgate.net/publication/372524254_Towards_the_Construction_of_a_Software_Benchmarking_Dataset_via_Systematic_Literature_Review