A Framework for Microservice Organizational Structure Optimization

Li, Xiaozhou; Albano, Michele

Published in:
SATrends '24: Proceedings of the 1st International Workshop on New Trends in Software Architecture

DOI (link to publication from Publisher):
10.1145/3643657.3643913

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

A Framework for Microservice Organizational Structure Optimization

Xiaozhou Li
xiaozhou.li@oulu.fi
University of Oulu
Oulu, Finland

Michele Albano
mialb@cs.aau.dk
Aalborg University
Aalborg, Denmark

ABSTRACT

With the soaring popularity of microservices, practitioners have realized that the key to success lies more in the management of people than in the architecture itself. Due to the systems' inevitable evolution, the organization shifts towards such changes where a lack of proper optimization can result in dreadful efficiency, error proneness, and toxic collaboration environments. However, despite the criticality, limited studies have proposed concrete support for optimizing microservice organizational structure through system evolution. This short paper sketches the vision of a framework to facilitate the optimization of microservice organizational structure. With this work, we aim to attract the attention of the microservice community toward the benefits and issues of the microservice organizational structure and propose a promising direction to a continuous solution.

KEYWORDS

microservice, organizational structure, optimization, gamification, framework

ACM Reference Format:
Xiaozhou Li and Michele Albano. 2024. A Framework for Microservice Organizational Structure Optimization. In *2024 International Workshop New Trends in Software Architecture (SATrends '24), April 14, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3643657.3643913

1 BACKGROUND AND MOTIVATION

During the last decade, microservice architecture has gained tremendous popularity in both industry and academia. Accompanied by the advance of its theory and practice on the technical level, the organizational structure of microservice remains unexplored and is, to a large extent, challenging to manage properly and a major barrier to effective microservice adoption [5]. It is commonly known that well-functioning project organizations can enable effective communication and collaboration, which simply lead to the success of the projects[7]. For practitioners, autonomous teams are critical to the success of microservice adoption, especially when many of the best practices largely accelerate the development and operation process [18]. For example, each team can make changes without permission from people outside the team or dependence on them, complete the work without communicating between different teams, and do most of the demanded testing without an integrated test environment.

However, it is difficult to maintain its performance, especially when the organizational structure is supposed to evolve like the evolution of the software architecture and fails to do so [10]. Meanwhile, the evolution of the organizational structure, including the growing collaborations between developers across the service boundaries, also largely influences the behaviors of the individual developers in terms of their performance and motivation [11]. Overlooking such situations forms a vicious circle of degradation in both software quality and developer morality. An infamous example of a system (though not microservice) failure due to suboptimal organizational structure is Windows Vista [32]. On the contrary, Amazon and Netflix are successful examples of clearly defined autonomous teams and well-maintained organizational structures [33].

Several works have been conducted towards providing methods to analyze the developers' collaboration in microservice or modular systems [22]. However, many studies are proposing approaches to analyze developer collaboration behaviors from a social science perspective using survey data, while several propose using GitHub data to analyze the developers' interaction with data-driven methods. Li et al. use social network analysis methods to reconstruct the collaboration network of microservice project developers and use network modularity as a metric to evaluate the organizational structure [23]. However, the study does not propose solutions to improve the quality of sub-optimal organizational structure. d'Aragona et al. find it is uncommon to have ideal distributed organizational structures for open-source microservice projects where one microservice is managed only by one developer or team [2]. However, the study does not propose potential solutions either.

One of the crucial issues for the degradation of microservice architecture is coupling [14], while such an issue remains regarding the organizational structures of microservices. "high cohesion low coupling" has long been a well-recognized software design principle, while, in particular, microservice, as one of the most dominantly popular modularized architectures, is also required to comply with the principle [41]. Regarding the coupling between actors in organizations in general, Beekun and Glick propose a method to measure from multiple dimensions, though the work is not specified for the application in microservice projects [6]. Regarding measuring the coupling in terms of the organizational structure of microservice projects, Li et al. propose the metric of organizational coupling using collective cross-service contribution weighted by developers' service-switching frequency [24]. However, the study does not propose solutions for decoupling.

SATrends '24, April 14, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0560-1/24/04
https://doi.org/10.1145/3643657.3643913

Regarding solving the coupling issues, as well as other anti-patterns in microservice projects, reconstruction of the target architectures is the critical first step. Many methods and tools have been proposed to facilitate the microservice architecture reconstruction [4]. Together with the reconstruction results, visualization of the architecture and the anti-patterns therein is also important [9]. However, limited studies have proposed support for reconstructing and visualizing the organizational structure and the organizational anti-patterns.

Towards solutions for optimizing the degrading organizational structure of microservice, besides the notification of anti-patterns via visualization, actionable approaches to mitigate such anti-patterns are required. Especially considering the fact that the core of organizational structure optimization is the management of developers, it is reasonable to adopt motivational designs to encourage positive actions. Gamification design, defined as the use of game design elements in non-game contexts, is a commonly adopted strategy that can provide motivational affordance [13]. Many studies have proposed using gamification design in software engineering [36]. It would also be useful to adopt specialized gamification design towards motivating developers into positive actions that avoid target anti-patterns in the organization.

Therefore, it is noticeable that the organizational structure of a microservice project has a huge impact on the quality of the architecture itself when the inevitable co-evolution of the architecture and the organizational structure can simply introduce potential anti-patterns. Due to the limited studies on such a research niche, this short paper proposes a framework and vision for optimizing the microservice organizational structure. The remainder of the article includes a brief introduction of the framework in Section 2, a discussion of the opportunities and challenges in Section 3, and a conclusion in Section 4.

2 ORGANIZATIONAL STRUCTURE OPTIMIZATION FRAMEWORK

The key activities toward organizational structure optimization shall include the following: 1) auto-logging, 2) data preprocessing, 3) continuous model training, 4) visualization, 5) recommendation, and 6) gamification (shown in Figure 1).

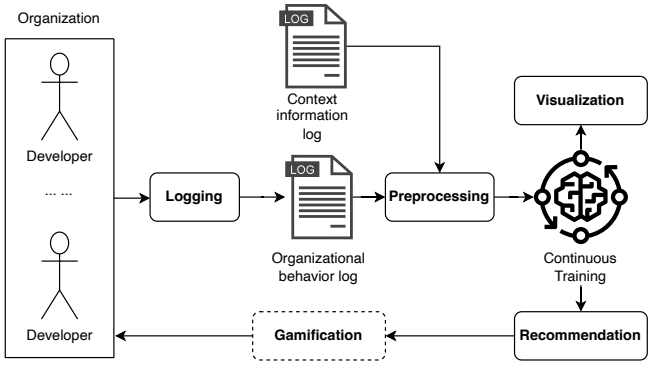

Figure 1: Organizational Structure Optimization Process

Auto-logging. The organizational structure optimization shall start with efficient organizational behavior logging, where detailed inter-organizational activities are recorded massively. Certain types of developers' behaviors, e.g., committing, issue addressing, commenting, etc., can be automatically retrieved from collaborative version control platforms, e.g., GitHub [27]. However, the difficulty lies where other organizational behavior data are required for future optimization, e.g., ways of interaction, innovation capability, competitiveness, etc. [40]. Obtaining information on these perspectives requires an additional step of analysis based on existing data types that can be automatically or manually logged. Meanwhile, a mapping between loggable behaviors and latent factors is critical, where analysis of survey data can be applied using factor analysis methods [16].

Preprocessing. The outcome of the logging step shall contain structured or unstructured data in various forms. For example, developers' contribution data can be recorded in JSON or CSV format, while developers' comments can be recorded as texts, including images. The obtained raw data must be preprocessed properly to perform analysis and prediction. The event log format proposed by Appice et al. [3] can be seen as the basic format for recording the organizational event, which can be enriched by adding extra data features.

Continuous Model Training. When the log data are sufficiently collected and properly preprocessed, they shall be sent to a set of pre-trained models, or pipelines, to provide useful predictive or verdictive results. The continuous data input and model training process shall follow the classic MLOps loop [31]. A good use case for such an MLOps pipeline in practice is the case of the OSSARA model, which can be used to predict the abandonment risk of open-source software continuously [26, 30]. Similar prediction activities shall include the ones targeting core developers' dropping out, developers' suitability to particular microservices, developers' collaboration suitability, organizational structure efficiency level, etc.

Visualization. Besides adopting AI models for predictive and verdictive purposes, visualization is also critical for stakeholders within the organization to easily perceive the trends and anomalies in the evolution of the organizational structure. For microservice projects, especially concerning their evolution of the source code and data flow perspectives, the combination of continuous architecture reconstruction and visualization has been widely studied on multiple layers [9, 17, 35]. The visualization layer of organizational structure can be integrated, while the organizational view shall help visualize the relationships within and identify the potential organizational anti-patterns therein [39].

Recommendation. To optimize the organizational structure of microservice projects, it is critical that the practitioners be aware of the issues and take action to solve them. Thus, a set of recommender mechanisms shall be designed based on individual developers' activities and collective organizational behaviors from the teams. Recommender systems in computer-supported collaboration environments can largely improve the efficiency of team communication and information handling [12]. Importantly, an implicit or explicit rating mechanism for relevant organizational activities and developers' preferences is required to apply a recommender system, while the "cold start" problem needs to be mitigated properly [38].

Gamification. As the recommender system provides recommendations on developers' organizational activities, the gamification design can be added as motivational affordances that enable developers to be more willing to take action. It is useful to adopt the basic elements of gamification design elements, e.g., point, badge, leaderboard, etc., in software projects [36]. It can also motivate developers towards more effective collaboration and contribution, but meanwhile less unnecessary couplings between service teams, as well as help to improve the project management process [28]. To be noted, the complexity of the applied gamification design varies greatly depending on the number of elements applied in terms of the mechanism, dynamics, and aesthetics [19]. Another layer to design more effective gamification is to target different developer profiles and typologies and provide personalized motivation affordances [25]. Notably, gamification design for organizational structure optimization is considered optional (marked with dotted lines in Figure 1).

In addition, the organizational structure optimization shall also take into account other context information recorded through other logging systems. The context information can include any external perspectives that may impact the organizational behaviors. For example, the enormous impact of COVID-19 on software projects and the developers' behaviors must be considered in order to cope with such a dire situation where both the productivity and the well-being of individuals must be taken care of [37].

3 OPPORTUNITIES AND CHALLENGES

The proposed framework for microservice organizational structure optimization shall contribute to the improvement in the human perspective of microservice architectures. The critical initial step of the actionable framework is to have a logging tool to collect organizational behavior data continuously. With the continuously accumulated and well-preprocessed data, an MLOps pipeline is implemented and run to provide in-time insights and recommendations, which shall greatly facilitate the optimization of the organizational structure for microservice projects targeting the enhancement of effective collaboration and the mitigation of unnecessarily coupled services and developers. The work to such an end can bring many opportunities but also result in challenges that might cause issues in the future.

3.1 Opportunities

Notably, the framework brings attention to the importance of the human aspect in microservice, which is one of the key factors of microservice success advocated by many practitioners but not sufficiently studied. For nearly a decade, the human factor has been indicated as pivotal to software engineering when such "soft subjects" are critical to software projects [8]. The maintenance of the highly dynamic architecture avoiding degradation depends heavily on the functioning of the teams and the organizational structure they form. The framework shall initiate the discussion in such a direction by proposing the possibility of obtaining organizational structure and developer behavior data via automated logging, with which future methods, techniques, and tools can be proposed and validated.

Furthermore, the framework includes adopting recommendation mechanisms to guide positive structure changes and using gamification designs to motivate developers to take positive actions. It is an opportunity for interdisciplinary studies on many relevant domains, e.g., data science, software engineering, game studies, etc. Studies herein shall also help social scientists understand the influence of motivational designs and recommendation mechanisms in changing human behavior within a given organization and such influences on the system quality. To the microservice community, it is valuable to validate the effectiveness of such designs adopted to maintain developers' positive behaviors, correct their wrong-doings, and furthermore maintain the overall system architecture quality from degrading. It shall lay the foundation for future enhancement in such designs.

It would also be interesting to see if the prevailing large language models and ChatGPT can facilitate the implementation of the framework. Studies have been conducted regarding the application of ChatGPT in creating basic microservice architecture [1] and crafting gamification designs for software engineering tasks [15]. There is an opportunity that such cutting-edge techniques can also be applied to support the practices suggested by the framework and enrich its repertoire.

3.2 Challenges

One of the challenges for implementing the framework is logging the behavior data. Provided the behavior data of the developers are automatically collected via tracing tools installed on their working devices, it shall require their consent to allow such data to be logged. It is, therefore, hazardous that such consents are not granted. An alternative approach to obtaining data is via questionnaires and interviews, which are time-consuming and require careful design. Another potential solution is to use online user-generated data when it also results in unexpected personal information leak [21]. Furthermore, such a data collection approach is more useful for geographically distributed open-sourced projects when anonymity can hinder proactive solutions from being applied to the target stakeholders. In addition, it is also challenging to validate the performance of such a framework, even partially, in a real-life microservice project from the industry due to confidentiality.

Furthermore, the potential ethical issues for adopting recommender systems and those for adopting gamification design in the context of software engineering should be paid extra attention. For example, the user privacy issue can also be problematic for recommender systems, together with potential anonymity breaches, behavior manipulation and bias, etc [29]. Solutions such as user-centered designs can be considered an adjustable tool to mitigate such issues [34]. Meanwhile, the potential physical and psychological harms caused by exploiting and manipulating user behavior by gamification shall also be considered before its adoption [20].

4 CONCLUSION

This short paper proposes a visionary framework for optimizing the organizational structures of microservice projects. The framework contains an MLOps pipeline that continuously learns from the organizational behavior data with a recommender system to provide positive action points to the target developers. Meanwhile,

the relevant stats and trends are visualized with anti-patterns notified to the stakeholders supporting decision-making. Furthermore, the framework also contains gamification design as a key process to provide motivational affordance to encourage developers to take action. By integrating all the mechanisms, the framework shall help the microservice teams pay constant attention to the changes in organizational structure, taking recommendations as actionable points while also having the motivation to accomplish, driven by gamification. The framework is the roadmap towards optimizing microservice architecture on the layer of humans in the organization.

ACKNOWLEDGMENTS

This work is based upon work supported by a grant from the Research Council of Finland (grant n. 349488 - MuFAno).

REFERENCES

[1] Aakash Ahmad, Muhammad Waseem, Peng Liang, Mahdi Fahmideh, Mst Shamima Aktar, and Tommi Mikkonen. 2023. Towards human-bot collaborative software architecting with chatgpt. In *EASE 2023*. 279–285.

[2] Dario Amoroso d'Aragona, Xiaozhou Li, Tomas Cerny, Andrea Janes, Valentina Lenarduzzi, and Davide Taibi. 2023. One Microservice per Developer: Is This the Trend in OSS?. In *European Conference on Service-Oriented and Cloud Computing*. Springer, 19–34.

[3] Annalisa Appice, Marco Di Pietro, Claudio Greco, and Donato Malerba. 2016. Discovering and tracking organizational structures in event logs. In *New Frontiers in Mining Complex Patterns: 4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Porto, Portugal, September 7, 2015, Revised Selected Papers 4*. Springer, 46–60.

[4] Alexander Bakhtin, Xiaozhou Li, Jacopo Soldani, Antonio Brogi, Tomas Cerny, and Davide Taibi. 2023. Tools Reconstructing Microservice Architecture: A Systematic Mapping Study. *Agility with Microservices Programming, co-located with ECSA* 2023 (2023).

[5] Saša Baškarada, Vivian Nguyen, and Andy Koronios. 2018. Architecting microservices: Practical opportunities and challenges. *Journal of Computer Information Systems* (2018).

[6] Rafik I Beekun and William H Glick. 2001. Organization structure from a loose coupling perspective: A multidimensional approach. *Decision sciences* 32, 2 (2001), 227–250.

[7] Frederick P Brooks Jr. 1995. *The mythical man-month: essays on software engineering*. Pearson Education.

[8] Luiz Fernando Capretz. 2014. Bringing the human factor to software engineering. *IEEE software* 31, 2 (2014), 104–104.

[9] Tomas Cerny, Amr S Abdelfattah, Vincent Bushong, Abdullah Al Maruf, and Davide Taibi. 2022. Microservice architecture reconstruction and visualization techniques: A review. In *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 39–48.

[10] Melvin E Conway. 1968. How do committees invent. *Datamation* 14, 4 (1968), 28–31.

[11] Robertas Damaševičius. 2010. On the human, organizational, and technical aspects of software development and analysis. *Information systems development: Towards a service provision society* (2010), 11–19.

[12] Ernesto Damiani, Paolo Ceravolo, Fulvio Frati, Valerio Bellandi, Ronald Maier, Isabella Seeber, and Gabriela Waldhart. 2015. Applying recommender systems in collaboration environments. *Computers in Human Behavior* 51 (2015), 1124–1133.

[13] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 9–15.

[14] Dario Amoroso d'Aragona, Luca Pascarella, Andrea Janes, Valentina Lenarduzzi, and Davide Taibi. 2023. Microservice logical coupling: A preliminary validation. In *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 81–85.

[15] Tommaso Fulcini and Marco Torchiano. 2023. Is ChatGPT Capable of Crafting Gamification Strategies for Software Engineering Tasks?. In *Proceedings of the 2nd International Workshop on Gamification in Software Development, Verification, and Validation*. 22–28.

[16] Richard L Gorsuch. 2014. *Factor analysis: Classic edition*. Routledge.

[17] Mia E Gortney, Patrick E Harris, Tomas Cerny, Abdullah Al Maruf, Miroslav Bures, Davide Taibi, and Pavel Tisnovsky. 2022. Visualizing microservice architecture in the dynamic perspective: A systematic mapping study. *IEEE Access* (2022).

[18] Jez Humble and Gene Kim. 2018. *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations*. IT Revolution.

[19] Robin Hunicke, Marc LeBlanc, Robert Zubek, et al. 2004. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, Vol. 4. San Jose, CA, 1722.

[20] Tae Wan Kim and Kevin Werbach. 2016. More than just a game: ethical issues in gamification. *Ethics and Information Technology* 18, 2 (2016), 157–173.

[21] Aleksandra K Krotoski. 2012. Data-driven research: open data opportunities for growing knowledge, and ethical issues that arise. *Insights: the UKSG journal* 25, 1 (2012), 28–32.

[22] Xiaozhou Li, Amr S Abdelfattah, Ruoyu Su, Joseph Lee, Ernesto Aponte, Rachel Koerner, Tomas Cerny, and Davide Taibi. 2023. Metrics and Models for Developer Collaboration Analysis in Microservice-Based Systems. A Systematic Mapping Study. In *17th International Conference on Software Process and Product Measurement (Mensura)*.

[23] Xiaozhou Li, Amr S Abdelfattah, Jorge Yero, Dario Amoroso d'Aragona, Tomas Cerny, and Davide Taibi. 2023. Analyzing Organizational Structure of Microservice Projects based on Contributor Collaboration. In *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 1–8.

[24] Xiaozhou Li, Dario Amoroso d'Aragona, and Davide Taibi. 2023. Evaluating Microservice Organizational Coupling Based on Cross-Service Contribution. In *International Conference on Product-Focused Software Process Improvement*. Springer, 435–450.

[25] Xiaozhou Li, Chien Lu, Jaakko Peltonen, and Zheying Zhang. 2019. A statistical analysis of Steam user profiles towards personalized gamification. In *3rd International GamiFIN Conference, GamiFIN 2019*. CEUR-WS.

[26] Xiaozhou Li, Sergio Moreschini, Fabiano Pecorelli, and Davide Taibi. 2022. OSSARA: abandonment risk assessment for embedded open source components. *IEEE Software* 39, 4 (2022), 48–53.

[27] Xiaozhou Li, Sergio Moreschini, Zheying Zhang, and Davide Taibi. 2022. Exploring factors and metrics to select open source software components for integration: An empirical study. *Journal of Systems and Software* 188 (2022), 111255.

[28] Liliana Machuca-Villegas and Gloria Piedad Gasca-Hurtado. 2019. Gamification for improving software project management processes: a systematic literature review. In *CIMPS 2018)*. Springer, 41–54.

[29] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. 2020. Recommender systems and their ethical challenges. *Ai & Society* 35 (2020), 957–967.

[30] Sergio Moreschini, David Hästbacka, and Davide Taibi. 2023. MLOps Pipeline Development: The OSSARA Use Case. In *Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems*. 1–8.

[31] Sergio Moreschini, Francesco Lomio, David Hästbacka, and Davide Taibi. 2022. MLOps for evolvable AI intensive software systems. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 1293–1294.

[32] Nachiappan Nagappan, Brendan Murphy, and Victor Basili. 2008. The influence of organizational structure on software quality: an empirical case study. In *Proceedings of the 30th international conference on Software engineering*. 521–530.

[33] Sam Newman. 2021. *Building microservices*. " O'Reilly Media, Inc.".

[34] Dimitris Paraschakis. 2017. Towards an ethical recommendation framework. In *2017 11th international conference on research challenges in information science (RCIS)*. IEEE, 211–220.

[35] Garrett Parker, Samuel Kim, Abdullah Al Maruf, Tomas Cerny, Karel Frajtak, Pavel Tisnovsky, and Davide Taibi. 2023. Visualizing Anti-Patterns in Microservices at Runtime: A Systematic Mapping Study. *IEEE Access* (2023).

[36] Oscar Pedreira, Félix García, Nieves Brisaboa, and Mario Piattini. 2015. Gamification in software engineering–A systematic mapping. *Information and software technology* 57 (2015), 157–168.

[37] Paul Ralph, Sebastian Baltes, Gianisa Adisaputri, Richard Torkar, Vladimir Kovalenko, Marcos Kalinowski, Nicole Novielli, Shin Yoo, Xavier Devroey, Xin Tan, et al. 2020. Pandemic programming: How COVID-19 affects software developers and how their organizations can help. *Empirical software engineering* 25 (2020), 4927–4961.

[38] Badrul M Sarwar, Joseph A Konstan, Al Borchers, Jon Herlocker, Brad Miller, and John Riedl. 1998. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. 345–354.

[39] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl. 2020. Microservices antipatterns: A taxonomy. *Microservices: Science and Engineering* (2020), 111–128.

[40] Quangyen Tran and Yezhuang Tian. 2013. Organizational structure: Influencing factors and impact on a firm. (2013).

[41] Priyal Walpita. 2020. Coupling and Cohesion in Microservices. https://priyalwalpita.medium.com/coupling-and-cohesion-in-microservices-235ed9203843.