Enhancing System Quality Attributes via Microservices Adoption

Roberta Capuano[1]

[1]University of L'Aquila, L'Aquila 67100, Italy

Abstract

In the last decade, industries and the research community have increasingly focused their attention on systems migration. System migration aims to modernize existing monolithic systems into modern and more scalable microservices-based architectures. Such migration consists of two phases: architectural refactoring and system deployment. Unfortunately, quality attributes did not receive enough attention during the architectural refactoring phase. Nevertheless, migration can be an instrument to improve the quality of the system. In this research, we propose a process that aims to enhance a set of system quality attributes during the system migrating to microservices. The proposed approach helps software architects to fulfill non-functional requirements qualities during the migration process.

Keywords

system migration, system modernization, monolithic architecture, microservices architecture, quality attributes, architectural refactoring

1. Introduction and Motivations

Nowadays, millions of applications are based on monolithic systems [1]. In the last decade, industries have started to migrate their systems into modern microservices-based architecture. The motivation for such evolution is to keep up with modern technologies and to maintain better business value. At the same time, the research community started studying this new trend trying to support companies in this challenge [2]. System migration refers to the process of refactoring an existing software and is composed of two phases: *architectural refactoring* and *system deployment*. The *architectural refactoring* consists of changing the structure of the system without altering its features and it is composed of i) comprehension of the system, and ii) microservices identification [3]. Unfortunately, in the *architectural refactoring* phase, functional and non-functional requirements can differ from the original specification [4]. The *system deployment* consists of microservices packaging using a containerization technology, e.g., Docker. The state of the art propose different sets of attributes for assessing the quality of the system after migration [5][6][7]. However, none of them put enough attention to quality attributes in the architectural refactoring phase. This research project aims to define a process to enhance a set of system quality attributes using migration to microservices. The following research questions will guide our goal:

- **RQ1** How to guide the architectural refactoring phase by using quality attributes?

ECSA'21: Doctoral Symposium, September 13–17, 2021, Växjö, Sweden
roberta.capuano@graduate.univaq.it (R. Capuano)

- **RQ2** How to chose the proper microservices identification approach considering a set of quality attributes in the migration to microservices scenario?
- **RQ3** How to find a tradeoff to combine quality attributes in the decomposition of the system into microservices?

2. Related work

In the literature, different researches address the problem of migration to microservices from monolithic systems. In [8], the authors present a program analysis-based method to migrate monolithic applications to a microservices architecture using static and dynamic analysis of the system. Their method has high accuracy and low performance cost. The authors in [3], propose an approach for automatic identification of microservices from Object Oriented source code, measuring both the structural and behavioural validity of the identified microservices and their data autonomy. In [9], a tool to continuously streaming down performance data, analyzing them and feeding back to the migration process the results of the analysis is presented. This approach guarantee that the new system does not fall short in terms of performance. The authors in [10] consider five quality criteria observed as relevant by practitioners for system migration. Compared with a baseline approach that considers just coupling and cohesion, their approach reinforced the need for adopting more criteria than traditional ones. In [11], the authors decompose a monolith application into independent microservices. The functionality distribution is optimized to guarantee a system with high cohesion, low coupling and good-sized services focused on core functionality. The authors in [12] propose the Functionality-oriented Service Candidate Identification (FoSCI) framework for the identification of service candidates from a monolithic system. Both the dynamic analysis and search-based functional atom grouping algorithm serves to service candidates extraction. The service candidate evaluation suite uses 8 different metrics, measuring functionality, modularity, and evolvability of the identified service candidates. In [13] the authors provide a validation framework for microservices resulted from (semi-)automatic decomposition of monoliths validated through an open-source framework.

3. Proposed Approach

The goal of this research project is to define a process for system quality improvement by migrating to microservices. This process will also guide software architects migrating their systems to microservices while preserving or achieving predefined non-functional requirements. The proposed approach matches the common system migration to microservices process focusing on a set of system quality attributes. Below, the proposed process shown in Figure 1 is described.

- **P1: System's Comprehension**. This phase includes all the activities aiming to understand the system by using different artefacts: models, code, traces, and logs. In our process, this stage is quality-driven and matches the static and dynamic analysis carried-out by using the above-mentioned artefacts. Since architects may or may not know the quality aspects to improve, a preliminary system analysis is required. The resulting set of quality attributes is the baseline for the migration. For each identified quality attribute, we provide a set of architectural anti-patterns supporting architects in recognizing the elements

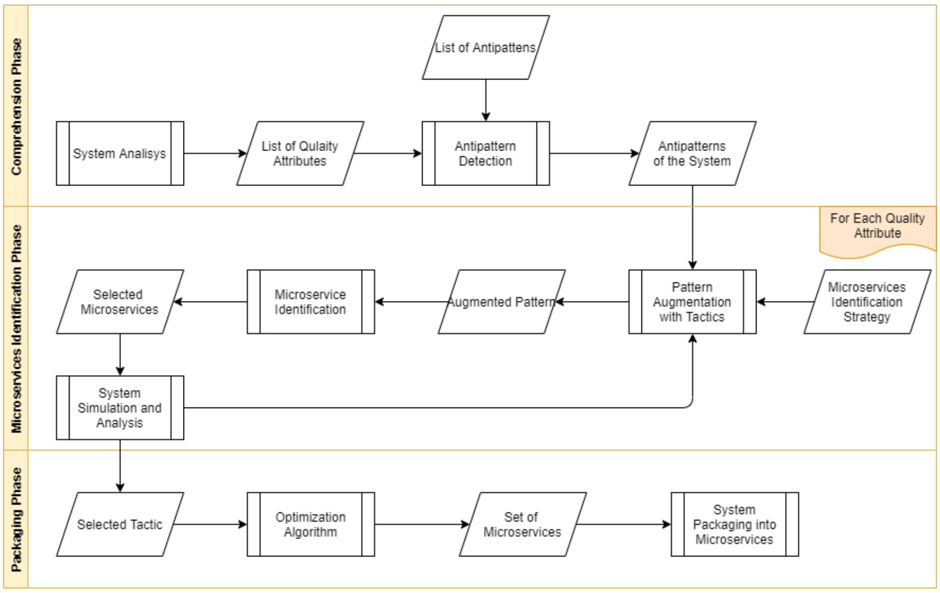

Figure 1: Proposed Process.

degrading the systems quality. This core operation is called anti-patterns detection and its output is the instances number of the specified anti-patterns and a set of related patterns.

- **P2: Microservices Identification**: this stage is quality driven since it directly inherits the quality aspects by the system's comprehension phase. **P1** result is a list of patterns aiming to solve the associated anti-patterns. A pattern consists of several tactics [14]. Our goal is to augment each pattern with a tactic consisting of a well-defined microservices identification strategy based on a static or dynamic view of the system. Consequently, the microservices identification is performed according to the augmented pattern. The new system is simulated and analysed to check if the quality attribute is achieved. These steps are repeated until the most suitable tactic is added to the pattern. The microservices identification phase gives as output one possible decomposition for each quality attributes to be improved.
- **P3: Microservices Packaging**: from **P2** we inherited a collection of possible decomposition where each of them refers to a single quality attribute to be improved. Since our goal is to suggest the optimal system decomposition strategy according with a set of quality attributes, this phase needs a pre-processing step. To this end, we will provide a multi-objective optimization algorithm to find the optimal decomposition. The output of this algorithm will be the set of microservices to be packaged and deployed.

4. Proposed Research Plan

- **S1: Systematic Literature Review on migration to microservice considering quality attributes aspects.** This step focuses on the identification of microservices considering quality attributes. The result will be a set of quality attributes that forms the baseline for the next steps. Moreover, it will allow us to identify, for each quality attribute considered, a set of microservices identification approaches. The tactics provided by our approach will be built on top of this collection. This phase answers to **RQ1** and **RQ2**.

- **S2: Review, collection and categorization of architectural anti-patterns degrading systems quality.** While investigating both research works and industrial perspectives, this step considers the set of quality attributes retrieved previously. The output will be a set of anti-patterns for each quality attribute identified in **S1** and their counterpart.
- **S3: Definition of tactics for quality improvement.** We define a set of tactics for each pattern classified in **S2**. Those tactics consist of a collection of microservices identification strategies. The results of **RQ2** will guide the outline of this step. Moreover, **S1** and **S2** of this research plan will answer to **RQ1**.
- **S4: Implementation of the multi-objective optimization algorithm.** Our process allows to find the optimal system decomposition according to a predefined set of quality attributes. Thus, after performing a review, the chosen optimization algorithm will be implemented. It will take into consideration both the selected tactics and the decomposed microservices. This step answer to **RQ3**.
- **S5: Evaluation of results.** Finally, the process will be applied on a real-world applications. This phase will serve to assess the validity of the proposed process.

We are currently working on **S1** and **S5**. For **S1** following the Systematic Literature Review Protocol defined in [15], we carried-out the planning phase setting up i) the research questions, ii) the search string, iii) the digital libraries for the search and iv) inclusion and exclusion criteria. Currently, we are involved in the data extraction and monitory phase.For **S5**, we are collaborating with a national company that is migrating their monolithic system to microservices by considering mainly performance. Unfortunately, they did not considered performance during the architectural refactoring phase. Thus, our goal is to i) study the design decisions behind the microservices identification they carried-out ii) analyse the monolith according to a set of quality attributes concerted with the company including performance iii) use our process to suggest a different decomposition strategy to evaluate its validity.

Acknowledgments

The authors research is supervised by Prof. Henry Muccini. The research program is supported by the Italian PON (National Operational Programme on Research and Innovation).

References

[1] R. C. seacord, D. Plakosh, G. A. Lewis, Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices, Addison-Wesley Longman Publishing Co., Inc., 2003.

[2] P. D. Francesco, I. Malavolta, P. Lago, Research on architecting microservices: Trends, focus, and potential for industrial adoption, in: 2017 IEEE International Conference on Software Architecture (ICSA), 2017, pp. 21–30. doi:10.1109/ICSA.2017.24.

[3] A. Selmadji, A.-D. Seriai, H. Bouziane, C. Dony, R. Mahamane, Re-architecting oo software into microservices: A quality-centred approach, Lecture Notes in Computer Science (2018) 65–73. doi:10.1007/978-3-319-99819-0_5.

[4] O. Zimmermann, Architectural refactoring for the cloud: A decision-centric view on cloud migration (2017) 129–145. doi:10.1007/s00607-016-0520-y.

[5] M.-D. Cojocaru, A. Uta, A.-M. Oprescu, Attributes assessing the quality of microservices automatically decomposed from monolithic applications, in: 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC), 2019, pp. 84–93. doi:10.1109/ISPDC.2019.00021.

[6] S. Ghayyur, A. Razzaq, S. Ullah, S. Ahmed, Matrix clustering based migration of system application to microservices architecture, International Journal of Advanced Computer Science and Applications (2018) 284–296. doi:10.14569/IJACSA.2018.090139.

[7] N. Alshuqayran, N. Ali, R. Evans, A systematic mapping study in microservice architecture, in: 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 2016, pp. 44–51. doi:10.1109/SOCA.2016.15.

[8] Z. Ren, W. Wang, G. Wu, C. Gao, W. Chen, J. Wei, T. Huang, Migrating web applications from monolithic structure to microservices architecture, in: Proceedings of the Tenth Asia-Pacific Symposium on Internetware, Association for Computing Machinery, 2018. doi:10.1145/3275219.3275230.

[9] A. Janes, B. Russo, Automatic performance monitoring and regression testing during the transition from monolith to microservices, in: 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2019. doi:10.1109/ISSREW.2019.00067.

[10] L. Carvalho, A. Garcia, T. E. Colanzi, W. K. G. Assunção, M. J. Lima, B. Fonseca, M. a. Ribeiro, C. Lucena, Search-based many-criteria identification of microservices from legacy systems, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 2020. doi:10.1145/3377929.3390030.

[11] F.-D. Eyitemi, S. Reiff-Marganiec, System decomposition to optimize functionality distribution in microservices with rule based approach, in: 2020 IEEE International Conference on Service Oriented Systems Engineering (SOSE), 2020. doi:10.1109/SOSE49046.2020.00015.

[12] W. Jin, T. Liu, Y. Cai, R. Kazman, R. Mo, Q. Zheng, Service candidate identification from monolithic systems based on execution traces, IEEE Transactions on Software Engineering (2021). doi:10.1109/TSE.2019.2910531.

[13] M. Cojocaru, A. Uta, A.-M. Oprescu, Microvalid: A validation framework for automatically decomposed microservices, in: 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2019. doi:10.1109/CloudCom.2019.00023.

[14] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, 3rd ed., Addison-Wesley Professional, 2012.

[15] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, 2007.