CODE STYLE GUIDELINES BY LOW-CODE TECHNOLOGIES AND OUTSYSTEMS

Borislav Shumarov

University of Library Studies and Information Technologies

Abstract: This paper investigates the perceptions and experiences of individuals involved in the software development field, with a specific focus on low-code and OutSystems, regarding code style guidelines. Through the use of a questionnaire, data was collected and analyzed using descriptive statistics. The results revealed a general consensus among participants regarding the benefits of code style guidelines, along with variations in beliefs and experiences related to their implementation. The findings contribute to understanding the role and impact of code style guidelines in low-code software development.

Keywords: code style guidelines, low-code development, OutSystems, questionaire, descriptive statistics.

Introduction

Code style guidelines play a role in software development by providing a set of conventions and best practices for writing and structuring code. They serve as a reference for developers to ensure consistency, readability, and maintainability of the codebase. In recent years, the emergence of low-code development platforms, such as OutSystems, has brought new opportunities and challenges to the field of software development.

Low-code software development has gained significant attention in recent years, providing a rapid and efficient approach to application development. However, challenges related to user interface customization, implementation of business logic, and integration have been identified in the literature.

A recent literature review [1] highlights challenges related to incomplete or incorrect documentation and the lack of learning and guiding resources for the software development with low-code development platforms. Such challenges can impede the effective utilization of low-code platforms. Another study also finds, that a steep learning curve and lack of material is one of the challenges with low-code technologies' facilitation [2]. There are also examples [1] where the adoption of standards has proven successful in the low-code development domain. The introduction of general standards enables

interoperability and facilitates collaboration. In the context of low-code development, the proposal and application of standards and guidelines could provide guidance and improve the overall software development process both for developers, as well as surrounding (some of which non-technical) roles in the complex process of modern software development.

This study aims to explore the perceptions, experiences, and opinions of people involved with IT in any form, with a main focus on those, participating in the low-code software development field and OutSystems. The primary objective is to gather insights into the benefits, challenges, and effectiveness of code style guidelines in this unique software development domain. This is made as a means of tackling the problem with the lack of guidance and enough resources, when using low-code technologies, as well as to facilitate more efficient software development process within a team and across teams overall.

The methodology employed in this study involved the use of a questionnaire as the primary data collection instrument. Through the questionnaire, participants were asked to share their perceptions and experiences related to code style guidelines in the context of OutSystems development. The data collected were then analyzed using appropriate statistical techniques, including descriptive statistics, to identify patterns, trends, and common themes among the responses.

The results obtained from the analysis of the collected data shed light on the overall inclination and consensus of the participants toward code style guidelines in the low-code development environment. Additionally, the study explores variations in beliefs and experiences related to the implementation and maintenance of code style guidelines. By examining these aspects, this research aims to contribute to the understanding of the role and impact of code style guidelines in the context of low-code software development.

The subsequent sections of this paper present the methodology employed in the study, the results obtained from the data analysis, and the conclusions derived from the findings.

Methodology of the Study

The study utilized a questionnaire to gather data on Code Style Guidelines in Software Development, with a specific focus on the context of OutSystems. The questionnaire aimed to investigate participants' perceptions, experiences, and opinions regarding code style guidelines and their implementation in the low-code development environment.

The questionnaire was designed to capture relevant information related to code style guidelines. The design of the questionnaire was also influenced by the specific characteristics and requirements of the OutSystems low-code development platform.

This study aimed to focus mostly on participants who have experience in low-code development, specifically with the OutSystems platform, which is a narrow category in itself. The selection aimed to include not only developers, but a diverse range of professionals, such as architects, and project managers, among other to gather insights from various perspectives.

The data collection process involved distributing the questionnaire electronically to the selected participants. Clear instructions were provided to ensure a consistent understanding of the questions and to encourage meaningful responses. The participants were given an appropriate amount of time to complete the questionnaire, and reminders were sent to maximize the response rate.

The collected data was analyzed using appropriate statistical techniques, including descriptive statistics, to summarize the responses and identify patterns and trends. The quantitative data obtained from the questionnaire provided insights into participants' perceptions of the benefits, challenges, and effectiveness of code style guidelines in the context of OutSystems development.

Results

Exploratory data analysis has been performed to the data from the questionaire. It provides descriptive statistics that summarize the central tendency, dispersion, and shape of a datase's distribution. The rows (top to bottom) of the Table are as follows:

1. Count: Number of non-missing values in each column.
2. Mean: Arithmetic mean (average) of each column.
3. Standard Deviation: Measure of the spread of values around the mean.
4. Minimum: Smallest value in each column.
5. 25th Percentile (Q1): Value below which 25% of the data falls.
6. 50th Percentile (Median or Q2): Value below which 50% of the data falls.
7. 75th Percentile (Q3): Value below which 75% of the data falls.
8. Maximum: Largest value in each column.

Table 1. Descriptive statistics analysis

	count	mean	std	min	25%	50%	75%	max
I believe that following a code style guide is beneficial for the overall quality of a codebase.	27.00	4.44	0.89	1.00	4.00	5.00	5.00	5.00
I believe that following a code style guide helps in understanding the code, when refactoring or reading code, written by someone else.	27.00	4.48	0.98	1.00	4.00	5.00	5.00	5.00
If we start using a style guide, I believe, that maintaining it in written form would be beneficial.	2.00	3.00	2.83	1.00	2.00	3.00	4.00	5.00
If we start using a style guide, I believe, that maintaining it in written form would have a negative impact on the speed with which we develop software.	2.00	4.00	1.41	3.00	3.50	4.00	4.50	5.00
If we start using a style guide, I believe, that maintaining it in written form would be beneficial.	3.00	4.00	1.00	3.00	3.50	4.00	4.50	5.00
If we start using a style guide, I believe, that maintaining it in written form would have a negative	3.00	3.00	0.00	3.00	3.00	3.00	3.00	3.00

impact on the speed with which we develop software.								
I believe, that maintaining a style guide in written form would be beneficial.	10.00	4.50	0.53	4.00	4.00	4.50	5.00	5.00
I believe, that maintaining it in written form would have a negative impact on the speed with which we develop software.	10.00	2.10	0.88	1.00	2.00	2.00	2.00	4.00
I believe, that maintaining it in written form is beneficial.	12.00	4.00	1.04	1.00	4.00	4.00	4.25	5.00
I believe, that maintaining it in written form has a negative impact on the speed with which we develop software.	12.00	2.42	1.31	1.00	1.75	2.00	3.25	5.00
I feel that the code style guide is consistently enforced across the team.	12.00	3.50	1.17	1.00	3.00	4.00	4.00	5.00
I have had to spend a significant amount of time refactoring or enforcing rules to conform to the style guide.	12.00	3.00	1.28	1.00	2.00	3.00	4.00	5.00

Table 1 shows the results from the EDA: overall, respondents express a moderately strong inclination toward the benefits of adhering to a code style guide, as evidenced by the mean score of 4.44 out of 5. This suggests a general consensus among the participants regarding the

positive influence of code style guidelines on the overall quality of a codebase.

Furthermore, the respondents demonstrate a similar level of agreement when it comes to the belief that following a code style guide aids in understanding code written by others, particularly during refactoring or code reading situations. The mean score of 4.48 indicates a high degree of confidence in the ability of style guidelines to enhance code comprehensibility and maintainability.

Regarding the manner of maintaining a style guide, the responses reveal an interesting disparity. While a majority of respondents (75%) believe that maintaining the style guide in written form would be beneficial, a smaller proportion (25%) express concerns about its potential negative impact on the speed of software development. This discrepancy suggests that maintaining the style guide in written form could have varying implications across different teams or contexts.

Examining the experiences of the respondents, we find that a significant number have encountered challenges related to the code style guide. Approximately 58% of participants reported spending a substantial amount of time refactoring or enforcing rules to conform to the style guide. This indicates that some members of the team have had to invest additional effort to align with the guidelines, potentially impacting productivity and development timelines.

Moreover, around 41% of respondents admitted to experiencing difficulties in understanding the style guide. This highlights the importance of providing clear and concise documentation, as well as offering comprehensive training or support to ensure effective adoption and comprehension of the guidelines.

Additionally, a considerable portion of the respondents (42%) expressed the need to seek clarification or guidance on the code style guide, suggesting that clear communication and support systems are crucial for facilitating the implementation of the guidelines within the team.

Interestingly, a substantial number of participants (53%) also provided suggestions for improvements to the code style guide. This feedback highlights the iterative nature of style guides and the value of incorporating input from team members to enhance its effectiveness and relevance. It also shows, that the small Code Style Guide, provided by the low-code platform provider (OutSystems) is limited and could be built upon.

Particulary interesting are the answers to question “I think that having a code style guide, similar to these for traditional tech stacks (Java, Python, JavaScript, C#, C++, etc.) is appropriate for Outsystems

and other low-code technologies." (**Fig. 1**). 78% of respondents agree with the statements, highlighting the need of a standardized code style guideline for OutSystems and similar technologies, despite being low-code.

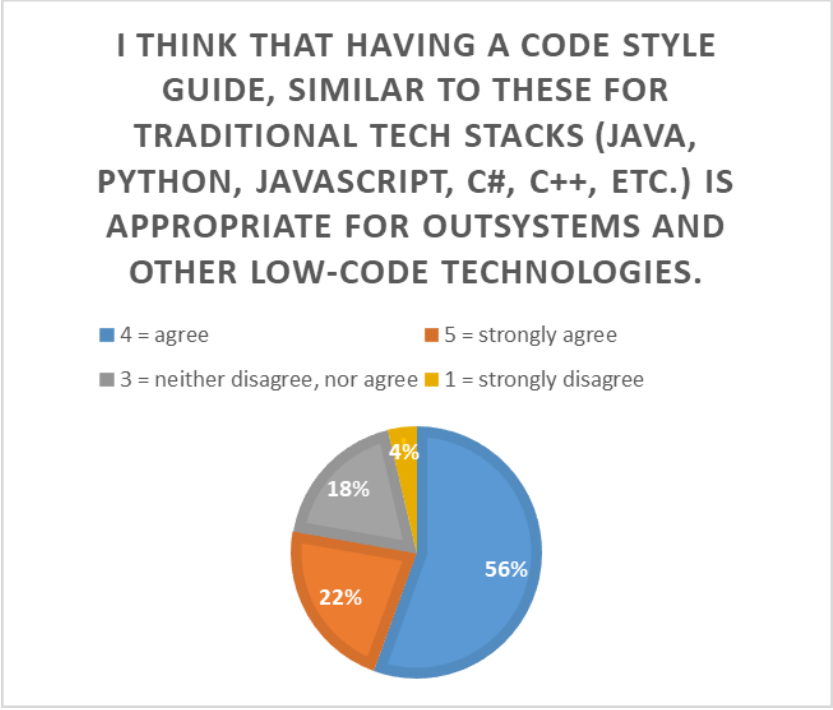

Fig. 1. I think that having a code style guide, similar to these for traditional tech stacks (Java, Python, JavaScript, C#, C++, etc.) is appropriate for Outsystems and other low-code technologies.

As to which areas are the most sought after to be built upon, results gathered for question "I have suggestions for additional rules to the Outsystems basic best practices in the area of (multiple choice):" highlight the following areas (**Fig. 2**):

- Naming
- Back-End Design Patterns
- Front-Ent Design Patterns
- Architecture
- Formatting

These results are especially valuable for the low-code and OutSystems community and suggest a gap in the current body of knowledge and practices in the specified areas.

Fig. 2. I have suggestions for additional rules to the Outsystems basic best practices in the area of (multiple choice)

Conclusions/Discussion

A survey has been performed among 38 participants with the help of a questionaire. The questions were focused on the area of Code Style Guidelines and OutSystems. In conclusion, the analysis of the gathered data suggest, that there is a perceived need for a broader and standardized best practices in the low-code software development area and the OutSystems community in particular in the form of a more extensive code style guideline.

There is a general consensus regarding the benefits of following a code style guide among the participants. However, variations exist in terms of the belief in maintaining the guide in written form and the

experiences related to its implementation. Understanding and addressing the challenges faced by team members, such as difficulties in comprehension or the need for guidance, can also contribute to the successful adoption and implementation of a code style guide. Additionally, leveraging the suggestions for improvements can foster continuous refinement and evolution of the style guide, leading to better code quality and streamlined development processes.

Благодарности: Тази статия е реализирана по проект на тема: „Модел на многосензорна система за откриване и съпровождане на превозни средства и дронове в умен град“, съгласно Договор № НИП-2023-06 от 28.04.2023 г.

References

1. **Rokis**, K., M. **Kirikova**, Challenges of Low-Code/No-Code Software Development: A Literature Review. // Perspectives in Business Informatics Research: 21st International Conference on Business Informatics Research, BIR 2022, Rostock, Germany, September 21 – 23, 2022, Proceedings. Cham: Springer International Publishing, 2022.
2. **Luo**, Y., P. **Liang**, C. **Wang**, M. **Shahin** & J. **Zhan**. Characteristics and challenges of low-code development: the practitioners' perspective. // Proceedings of the 15th ACM/IEEE international symposium on empirical software engineering and measurement (ESEM). 2021.

About the Author

Borislav Shumarov received his initial education in Bulgaria and then moved to pursue his studies in Germany, where he also started his career, while graduating from his bachelors and masters degree. Initially, he was involved with the economics & business field, but then decided to try out software development and liked it. So he is currently looking for ways to combine new, better technologies with existing business problems. He is passionate about quality code, system's architecture and design and everything low-code. His focus lies on Outsystems, but he is a generalist in the areas of full stack development on .Net, Java, SharePoint, and Mendix, which he has gathered during his years of experience as a software developer.

To contact the Author: shumarov.borislav@gmail.com

ИЗСЛЕДВАНЕ ВЪРХУ СТИЛ НА КОДА ПРИ НИСКОКОДОВИ ТЕХНОЛОГИИ И OUTSYSTEMS

Борислав Шумаров

Университет по библиотекознание и информационни технологии

Резюме: Тази статия изследва възприятията и опита на лица, занимаващи се с разработване на софтуер, с особено внимание върху нискокодови технологии и платформата OutSystems, относно ръководствата за стил на кода. Чрез използването на анкета са събрани данни, които са анализирани с помощта на описателна статистика. Резултатите показват общо съгласие сред участниците относно ползите от ръководствата за стил на кода. Проучването и анализът допринасят за изучаването на ролята и влиянието на ръководствата за стил на кода в разработването на софтуер с нискокодови технологии.

Ключови думи: ръководства за стил на кода, разработване с нискокодови технологии, OutSystems, анкета, описателна статистика.

За автора

Борислав Шумаров завършва своето средно образование в България, след което се премества, за да следва висше в Германия, където започва и своята кариера, докато завършва бакалавър и магистър. Първоначално се занимава с икономика и бизнес, но после решава да изпробва и разработката на софтуер, което му харесва. В момента търси начини да съчетае нови и по-добри технологии със съществуващи бизнес проблеми. Страстта му включва качествен код, архитектура и системен дизайн, както и всичко, свързано с нискокодови (low-code) технологии. Фокусът му е върху платформата OutSystems, но той е генералист в областта на разработката на .Net, Java, SharePoint и Mendix, които умения е придобил през годините си като софтуерен инженер.

За контакт с автора: shumarov.borislav@gmail.com