

Adoption and usability of low-code/no-code development tools

Tina Beranič, Patrik Rek, Marjan Heričko

Faculty of Electrical Engineering and Computer Science, University of Maribor

Koroška cesta 46, Maribor, Slovenia

{tina.beranic,patrik.rek,marjan.hericko}@um.si

Abstract. *The low-code/no-code development approach is an increasingly important domain addressing many current software development challenges. The available reports and forecasts point towards very positive global trends. Therefore, we looked into the popularity, adoption and use of low-code/no-code development approach in the Slovenian environment. This paper presents the results of a study made within business organizations revealing the adoption and acceptance of the low-code/no-code development approach and a usability study of the representative low-code/no-code development tool, expressed with SUS, UMUX and UMUX-Lite. The results reveal that the use of the low-code/no-code development approach in Slovenian organizations is lower than reported by global research. However, a significant overlap can be found in many areas, including reasons for not using the low-code/no-code development approach. According to the encouraging results of a usability study and the positive expectations related to the low-code/no-code development approach, positive changes are expected.*

Keywords. LCNC, use, adoption, prevalence, usability, companies, SUS, UMUX, UMUX-Lite

1 Introduction

Rapid changes have a significant impact on the development process itself and consequently also on company productivity and agility. The low-code/no-code development approach is an important concept addressing current challenges in the software development domain. Its origins date back to 1980s when rapid application development (RAD) tools were introduced as an alternative to traditional text-based development environments (Wong et al., 2019). However, its recent popularity can be attributed to the growing importance of digital transformation (Hecht, 2019; OutSystems, 2019; Bloomberg, 2017).

Many benefits to the low-code/no-code development approach can be found. Its use can speed up the development process and, consequently, lower costs (Ploder et al., 2019). Also, it reduces the connection to IT developers and allows for the integration of other departments in the development process (Wong et al., 2019; Ploder et al., 2019; Pantelimon et al., 2019). The low-code/no-code development approach is supported by a

variety of available platforms. According to Gartner's Magical Quadrant, visible representatives are the tools provided by Microsoft, Mendix, Salesforce, OutSystems and Appian (Vincent et al., 2019).

The forecast and assumptions about the use of low-code/no-code development platforms follow a positive curve. By 2024, more than 65% of applications will be developed using the low-code/no-code development approach and by the same year, more than 75% of large enterprises will use at least four low-code/no-code development tools (Wong et al., 2019). Even more, one of the available reports (OutSystems, 2019) indicates that 39% of the questioned organizations have already invested in the low-code/no-code development platform and 41% of the respondents answered that they have already implemented a low-code/no-code development approach.

While the above-mentioned forecasts and results apply on a global scale, the aim of this research was to look into the state of low-code/no-code development in Slovenia. Two complementary studies were implemented since the perceived positive usability can increase the intended use of the low-code/no-code development approach in the future. Therefore, the first study was aimed at researching the use of the low-code/no-code development approach in Slovenian organizations. A survey was conducted on the attitudes towards a low-code/no-code development and the adoption of available tools and platforms. Secondly, a usability study was done within the master students of the IT-related study program. Based on their practical experiences gathered in the study course, the usability of the used low-code/no-code tool was evaluated.

Consequently, our research followed the research questions:

- What is the attitude toward the low-code/no-code application development and related tools in Slovenian business organizations?
- How do master degree students evaluate the usability of a low-code/no-code development tool?

This paper is organized as follows. Section 2 presents the results of a study revealing the acceptance, adoption and popularity of low-code/no-code development in the Slovenian business environment, and Section 3 presents the implemented usability study among

master degree students. Finally, the paper is concluded in Section 4, where the gathered results are discussed and compared with related research.

2 Acceptance and adoption of low-code/no-code development in the business environment

Positive global trends connected with the use of low-code/no-code development, suggest that the use of low-code/no-code development platforms has also appeared among Slovenian business organizations. To gather current data, a survey questionnaire was distributed to potential participants in an online format, allowing for a greater reach and more convenient completion of the questionnaire. The structure of the questionnaire followed previously used questions by related global surveys. However, some, and in our opinion important questions, were added.

The invitation was answered by 78 participants within Slovenia, thus constituting a survey sample. The majority of respondents, 34.6%, were employed in small companies that have between 11 and 50 employees. 28.2% of the questioned participants were from companies with more than 250 employees, and 25.6% of the participants were employed in medium-sized companies, meaning from 51 to 250 employees. Only a minority, i.e. 11.5% of the participants were coming from companies with less than 10 employees. For almost 70% of the participants, software represents one of their business areas, followed by technology and the internet, chosen by 50% of participants, and consulting, selected by 34.6% of respondents. 9% of the companies were from the banking and finance sector, and 6.4% from the healthcare and pharmacy domain. 5.1% of respondents worked in insurance and 10.3% in energy.

26.9% companies employ between 50 and 100 people in the informatics department, among which, 47.6% of the companies have between 20 and 50 developers, and 28.6% companies have between 50 and 100 developers. On the other hand, 16.7% companies employ more than 100 people in the informatics department, and 19.3% employ less than 10 people in the department of informatics. 14.1% of the survey participants are CIO or CTO, and 5.1% occupy a management role not connected to IT. 10.3% of respondents were the head of IT within the company, and 25.6% had the role of a software architect. 7.7% of respondents were business analysts, 6.4% quality engineers and 53.8% of survey participants were developers.

The participants were asked about their familiarity and adoption with low-code/no-code development in their organizations. The answers are presented in Figure 1. 50% of management roles responded that they heard and know about the low-code/no-code development. The percentage is similar within techni-

cal roles, 43.6%. 25% of management respondents already tried low-code/no-code tools, but, only 10% already use low-code/no-code development approach. Within the technical roles, only 10.9% of survey participants already tried any low-code/no-code tools, while the percentage of use is similar within management roles, with 9.1%. The technical roles stand out with the answer that they did not hear about low-code/no-code approach before the questionnaire. That option was chosen by 36.4% of respondents.



Figure 1: Knowledge and use of low-code/no-code development approach and tools within management and technical roles.

We also asked the members of management if the low-code/no-code development approach presents a part of their IT strategy. 20% of the respondents answered that they already use or are planning to implement low-code/no-code development in the near future. In contrast, 60% do not intend to use the approach in their companies. Among those who already use low-code/no-code development, the approach was used by least five people. As they stated, the low-code/no-code tools are used by business users, in one case also by professional IT developers. In the company that is planning to start implementing low-code/no-code development, tools will be used by professional IT developers. Since the majority of respondents do not plan to introduce the low-code/no-code development approach, we researched the reasons. One of the most common reasons is the lack of knowledge about low-code/no-code development and its connected platform and tools. This answer was chosen by 68.3% of respondents. Other reasons are the concern for commitment to only one low-code/no-code tool provider, and doubts about the feasibility of the mentioned approach when building their applications. The latter reason was chosen by 75%. Some of them were also concerned about the scalability and security of the application developed using a low-code/no-code approach, with 43.8% and 37.5% of the selected answers, respectively.

The survey participants who already implement low-code/no-code approach or are planning its implemen-

Table 1: The opinion of the participants already using low-code/no-code development approach or planning to implement it in the near future.

	1	2	3	4	5	Average
With the approach, it is possible to develop web and mobile applications for use within a company.	0 %	7.7 %	7.7 %	61.5 %	23.1 %	4.0
With the approach, it is possible to develop web and mobile applications for customers outside the company.	7.7 %	15.4 %	7.7 %	61.5 %	7.7 %	3.5
The approach enables successful involvement and better cooperation with business users in the process of application development.	0 %	7.7 %	23.1 %	53.8 %	15.4 %	3.8
The approach can increase company agility.	0 %	7.7 %	15.4 %	46.2 %	30.8 %	4.0
The maintenance of products developed with the approach is easy.	15.4 %	0 %	30.8 %	46.2 %	7.7 %	3.3
The implementation of the approach increases the speed of the application's development.	0 %	15.4 %	0 %	61.5 %	23.1 %	3.9
Using the approach can save a lot of time.	0 %	0 %	7.7 %	76.9 %	15.4 %	4.1
Using the approach can free up developers' time to devote their time to more complex projects.	0 %	15.4 %	15.4 %	61.5 %	7.7 %	3.6
The approach is useful for automating repetitive development tasks, such as the development of input forms.	0 %	7.7 %	0 %	76.9 %	15.4 %	4.0
The approach is suitable for the development of mission-critical systems.	7.7 %	30.8 %	23.1 %	23.1 %	15.4 %	3.1
The approach can contribute significantly to the digital transformation of a company.	0 %	15.4 %	15.4 %	53.8 %	15.4 %	3.7
Using the approach, more time can be devoted to innovation and less to maintenance.	0 %	15.4 %	30.8 %	38.5 %	15.4 %	3.5

tation in the near future were asked to evaluate different statements connected to low-code/no-code development. Among all of the participants, 16.7% belong to this group. The results are presented in Table 1. According to the answers, they believe that the low-code/no-code approach is appropriate for the development of web and mobile applications that are going to be used within a company, with an average evaluation of 4.0 respectively. With a slightly lower score, the participants evaluated the option of using a low-code/no-code approach for the development of applications for their customers, with an average score of 3.5. According to the answers, the low-code/no-code approach is appropriate for automating repetitive tasks in the development process. However, it is slightly less appropriate for the development of mission-critical systems. With an average evaluation score of 4.1, the respondents agreed that the use of the low-code/no-code development approach could save a lot of time and could contribute to the digital transformation of a company, with an average score of 3.7.

Different no-code/low-code tools and platforms are available on the market, allowing for a rich choice among interested companies. Participants who have already implemented a low-code/no-code approach or are planning its implementation in the near future were asked to choose the tools they use or the tools they have tried and evaluated. The tools Lightning App Builder, Mendix, Appian and Microsoft Power Apps were chosen by 7.7% each and the tool Oracle APEX by 23.1%.

Participants in the management roles were also asked about the state of shadow IT within their company. Shadow IT represents the applications that are out of sight for the organization's IT personnel, pos-

sibly presenting an IT security and privacy risk (OutSystems, 2019; Gartner, Inc., 2020b). On the other hand, in citizen development, non-professional developers use the tools approved by IT, but, if not managed properly, they can present the same risk as shadow IT (OutSystems, 2019; Gartner, Inc., 2020a). 50% of the respondents were of the opinion that shadow IT does not present a risk for their organization, however, they are aware that it exists. A small portion, 10% of participants, perceive shadow IT in their company as a risk, while 40% were of the opinion that shadow IT does not exist in their company. A similar question looked into measurements taken to address challenges connected to shadow IT and citizen development. Only 10% answered that they had chosen the low-code/no-code tools and platforms that had to be used within citizen development in order to mitigate the possible risk. Others, either did not have an application developed by business users or their IT department does not manage shadow IT and citizen development, or tries to control those fields with limited resources.

According to positive expectations connected to low-code/no-code development detected in many available reports, the same question was also asked of Slovenian organizations. Among members of management, the opinion was perfectly split: 50% of the respondents believed that the positive expectations connected to low-code/no-code development will come true, and 50% disagreed. Among participants in technical roles, the opinion is slightly more pessimistic. 41.8% agreed with a positive outcome, while 58.2% did not share the positive view.

Consequently, their relationship to low-code/no-code development approach and related tools is ex-

pected. The results are presented in Figure 2. The most frequent answer among management and technical roles was sceptical, at 45% and 41.9% respectively. That emotion was followed by undecided, which accounted for 35% among management and 30.9% among technical roles. A positive attitude was found among 30% of those in management and by 29.1% in technical positions. However, only 10% of members of management and 18.2% of technical participants defined their attitude towards low-code/no-code approach as negative.

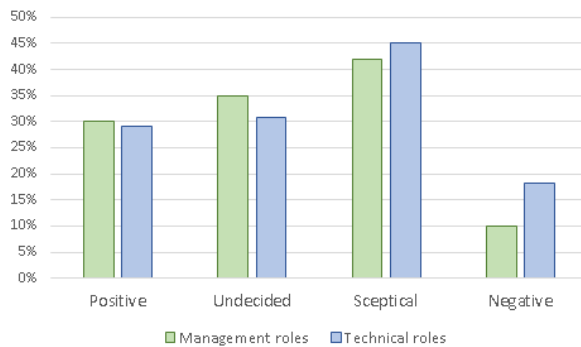


Figure 2: Participants' relationship and emotions with regard to low-code/no-code development, platforms and tools.

3 Usability of low-code/no-code tools by students

The second part of the presented research covers a usability study for the used low-code/no-code platform representative. Low-code/no-code platforms allow development to take place using a graphical interface instead of with a traditional text-based development environment (Waszkowski, 2019; Ploder et al., 2019; Hakimi, 2019). The tools can be used for private a single user and in-company applications to visualize data from various tables and sources. A practical example of an application developed with a representative low-code/no-code development tool is an expense tracker for a specific business department such as vehicle management. An implementation in Microsoft Power Apps platform is displayed in Figure 3. The application was automatically generated from a Microsoft Excel spreadsheet. It lists all the expenses for a company-owned car. The left-hand side of Figure 3 presents a details screen where the rating stars were manually added using a graphical editor, while the right-hand side of Figure 3 shows all expenses by the service provider. The graphical representation was added manually using the graphical editor in Microsoft Power Apps.

The Power Apps was used as a part of practical work conducted in the Informatics and Communication Technologies master study program. We used Mi-

crosoft Power Apps since it is included in the Microsoft Office 365 package used by the University of Maribor (Leung, 2017). The students' assignment was to develop a simple application in order to get familiar with the low-code/no-code development tool. According to the students' responses, and our practical experience with the example application, the main advantages of using a Power Apps tool can be summarized as:

- The interface is intuitive and easy to use.
- Automatic screen generation.
- Quick implementation of simple applications.
- Simple integration with a database.
- Instant application testing after each change.

As students have pointed out, the approach is appropriate for use as a minimally viable product for higher tier applications. The main disadvantages of the approach using Microsoft Power Apps can be summarized as:

- Unreliability in operation and common problems with Excel table connection.
- Weaker performance.
- Difficulty adding items that are not in the Microsoft Excel table.
- Unknown errors disappearing after restarting the application.
- Weak official documentation.
- Lower tool adoption, resulting in less support on the internet.
- No automatic saving of the work, resulting in lost work.

Following a hands-on approach, a usability study was conducted amongst 35 first grade master's degree students. A five point Likert scale was used for students to self-grade their knowledge of common programming languages (Joshi et al., 2015). The students evaluated the knowledge of the programming languages that they encountered during their studies. The average scores are displayed in Table 2. According to the results, students are most familiar with C# and JavaScript, which are both higher level languages. Consequently, we assumed that students would be familiar with low-code/no-code development tools since they represent an even higher level of abstraction from computer details.

Most of the students have at least two years of programming experience due to their undergraduate studies, while on average, they have four years of experience. 37.1% of students did not know low-code/no-code development before the study course, while 11.4% were familiar with low-code development or have used it before. The rest of the respondents had already heard of the mentioned tools but did not use them in practice. The results coincide with the data

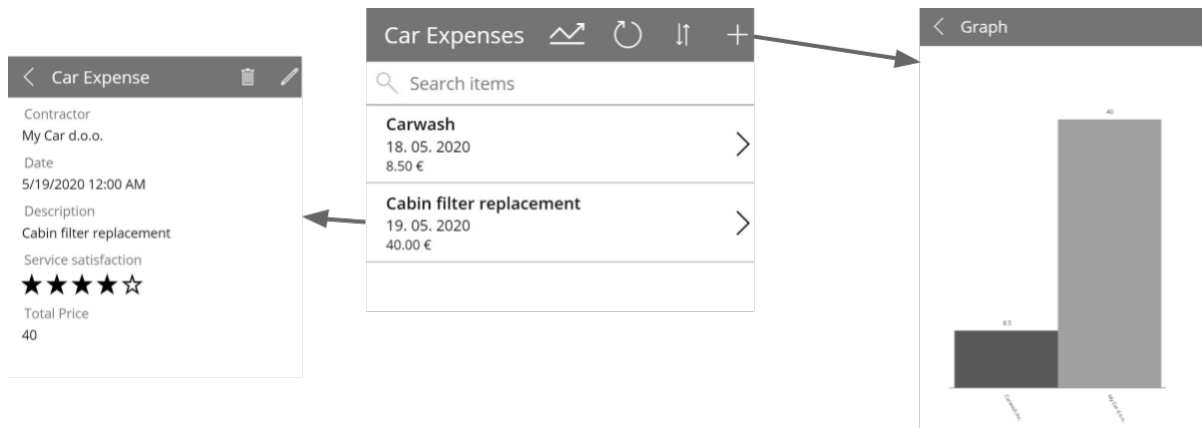


Figure 3: Example of an application developed in Microsoft Power Apps using the automatic generator from Microsoft Excel table.

Table 2: Average grades of students' programming knowledge.

	Average grade
Java	3.23
C++	2.57
C#	4.03
PHP	2.57
JavaScript	3.83
Python	2.83

about the use of the low-code/no-code tool in practice. The majority, or 87% of the students, answered that they never used low-code/no-code tools in business environments.

According to the low usage, which was also detected among a Slovenian business organizations, the usability of the used Power Apps tools was researched. The System usability scale (SUS) was used. The score can provide an insight into subjective perceptions of the usability of the evaluated system (Brooke, 2013). Students were asked to grade ten statements about the used tool answering with a five point Likert scale:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

We calculated the SUS value for each student according to their answers using the process described by Brooke et al. (1996). The maximum SUS score is 100, however, an average SUS result for Microsoft Power Apps tools was 65.64, with a standard deviation of 12.21 and median of 67.5 as displayed in Table 3. A different interpretation of the SUS score exists. According to Brooke (2013), a score of 65.64 can be explained as OK. It would, furthermore, receive a grade D and be on the higher margin of acceptability (Brooke, 2013). There is a slight negative correlation with a correlation coefficient of -0.283 between students' average skills in programming languages and SUS results, which means that the Microsoft Power Apps tool is evaluated as more usable for people with less experience in programming.

The usability metric for the user experience (UMUX) and its lighter successor UMUX-Lite was also obtained from surveyed students. UMUX is a quicker way to assess system usability with fewer questions for respondents. Compared to 10 questions in SUS, UMUX contains only four questions, where two of them are positive and two negative, measuring usability components' effectiveness, efficiency, and satisfaction (Finstad, 2010). Similarly, as SUS, it can score between 0 and 100 (Finstad, 2010). On the other hand, UMUX-Lite simplifies UMUX even more with only two questions (Lewis et al., 2013). Therefore, we used the following statements to obtain UMUX and UMUX-Lite results:

1. The system's capabilities meet my requirements.
2. Using this system is a frustrating experience.
3. I have to spend too much time correcting things with this system.
4. This system is easy to use.

For calculating UMUX-Lite, we have used the first

Table 3: Results of SUS, UMUX and UMUX-Lite usability metrics based on survey responses.

	Arithmetic mean	Median	Mode	Standard deviation	Correlation coefficient (average knowledge of programming languages)
SUS	65.64	67.5	65	12.21	-0.283
UMUX	66.43	68.75	75	16.71	-0.327
UMUX-Lite	67.24	71.65	71.65	11.55	-0.417

and fourth statement. As shown in Table 3, the calculated average score of UMUX is similar to the case of SUS, at 66.43. However, a slightly larger negative correlation coefficient on programming language knowledge can be detected, which furthermore supports the statement that the tool is more suitable for beginners in programming (Lewis et al., 2013; Finstad, 2010). As shown in Table 3, the average value of the metrics UMUX-Lite is 67.24. The graphical representation of all of the used usability metrics is presented in Figure 4.

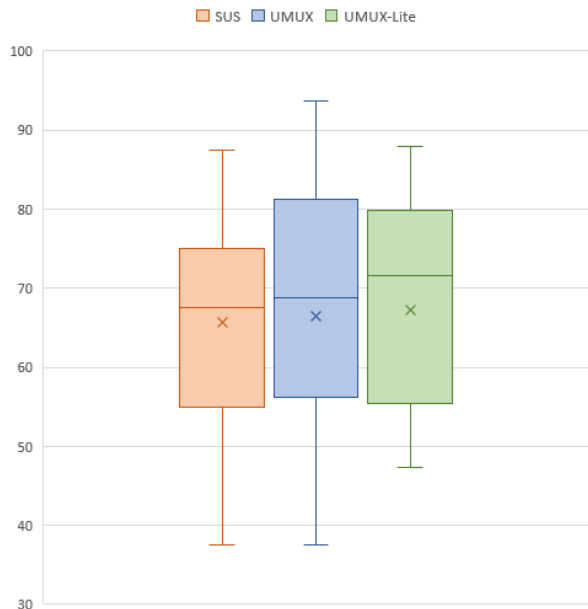


Figure 4: The graphical representation of the results of SUS, UMUX and UMUX-Lite.

At the end of the questionnaire, we also asked students about their expectations connected to low-code/no-code development. As detected within Slovenian business organizations, the current adoption of low-code/no-code development is rather low. However, a fair percentage of respondents believe in positive forecasts connected to low-code/no-code development. With an average score of 2.7, students' expectations are somehow pessimistic, not coinciding with answers from the business environment. But still, the portion of entirely negative responses is small, since more than 50% of the respondents answered with a neutral choice.

4 Discussion and conclusion

This paper presents the results of a study researching the use and adoption of low-code/no-code development in the Slovenian business organizations, and the usability study of a representative of low-code/no-code development tool implemented among master degree students. The survey was answered by 78 business participants and 35 students. Since the sample size was small, the results could not be generalized, but still offer a great insight into the current state of affairs.

According to the gathered data, only a small portion of organizations in Slovenia are already using a low-code/no-code development approach, just 10%. What is more, 60% of respondents said they do not plan to use the mentioned approach in the near future. By contrast, the results of global research (OutSystems, 2019) significantly differs, showing that 41% are already using the low-code/no-code development approach. The low adoption of the low-code/no-code development approach in the Slovenian environment can also be detected by the students' answers, since only 5.7% of students have already used or know low-code/no-code development tools.

The main reason that business organizations in Slovenia do not use the low-code/no-code development approach are doubts about the feasibility of the mentioned approach when building their applications, the lack of knowledge about low-code/no-code development and the connected platform and tools, and a concern for getting committed to only one low-code/no-code tool provider. The top three answers coincide with the results of global research (OutSystems, 2019), although the order differed. The most common response within Slovenian companies was the third place answer in global research. The agreement between Slovenian and global research can also be detected for the kinds of projects that low-code/no-code development is used for. Both put the applications that are used within the company in first place. Furthermore, the opinion of Slovenian respondents that the low-code/no-code approach is appropriate for automating receptive tasks is aligned with the results of a global survey (IDC, 2019). The numbers connected to shadow IT also align. 40% of the asked Slovenian business organizations were of the opinion that shadow IT does not exist in their company. The percentage in global research (OutSystems, 2019) is 36%. In Slovenia, 10% of participants perceive shadow IT in their company as a risk, which can be compared to a global percentage of 15%.

The second part of the presented study looked into the usability of the Microsoft Power Apps tool. The tool was used by master's degree students within a course assignment, wherein, in the end, the usability was evaluated. According to the results, the average SUS score was 65.64, which can be interpreted as OK. Similar results were also gathered from UMUX and UMUX-Lite, with 66.43 and 67.24, respectively. Since no related study measuring the usability of low-code/no-code development tools were detected, a comparison with related work could not be made.

Based on the gathered results and the comparisons made with the global survey, we can conclude that the use of the low-code/no-code development approach and connected tools in Slovenia negatively deviates from global results. However, since the respondents believe in positive expectations connected to low-code/no-code development, a growth in its use can be expected in the next few years.

Acknowledgments

The authors acknowledge financial support from the Slovenian Research Agency (research core funding No. P2-0057). Finally, the authors would like to express their appreciation to all survey participants for their valuable contributions.

References

- Bloomberg, J. (2017). *The Low-Code/No-Code Movement: More Disruptive Than You Realize*. Retrieved from <https://www.forbes.com/sites/jasonbloomberg/2017/07/20/the-low-codeno-code-movement-more-disruptive-than-you-realize> (Accessed on 07/05/2020)
- Brooke, J. (2013, 01). Sus: a retrospective. *Journal of Usability Studies*, 8, 29-40.
- Brooke, J., et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
- Finstad, K. (2010). The usability metric for user experience. *Interacting with Computers*, 22(5), 323-327.
- Gartner, Inc. (2020a). *Citizen Developer*. Retrieved from <https://www.gartner.com/en/information-technology/glossary/citizen-developer> (Accessed on 07/05/2020)
- Gartner, Inc. (2020b). *Shadow It*. Retrieved from <https://www.gartner.com/en/information-technology/glossary/shadow> (Accessed on 07/05/2020)
- Hakimi, E. (2019). *Low-code Development Platform*. Retrieved from <https://www.slideshare.net/EhsanHakimi/low-code-development-platform> (Accessed on 07/05/2020)
- Hecht, L. E. (2019). *Low-code platform adoption gets a boost from digital transformation*. <https://thenewstack.io/low-code-platform-adoption-gets-a-boost-from-digital-transformation/>.
- IDC. (2019). *The impact of low-code on it satisfaction* (Tech. Rep.). Appian. Retrieved from <https://www.appian.com/wp-content/uploads/2019/05/LowCode-and-IT-Satisfaction-Survey-Report.pdf>
- Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert scale: Explored and explained. *Current Journal of Applied Science and Technology*, 396-403.
- Leung, T. (2017). Introducing powerapps. In *Beginning powerapps* (pp. 3-14). Springer.
- Lewis, J. R., Utesch, B. S., & Maher, D. E. (2013). Umux-lite: when there's no time for the sus. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 2099-2102).
- OutSystems. (2019). *The state of application development, 2019/2020*. <https://www.outsystems.com/1/state-app-development-trends/>. Author.
- Pantelimon, S., Rogojanu, T., Braileanu, A., Stanciu, V., & Dobre, C. (2019). Towards a seamless integration of iot devices with iot platforms using a low-code approach. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (p. 566-571).
- Ploder, C., Bernsteiner, R., Schlögl, S., & Gschliesser, C. (2019). The future use of lowcode/nocode platforms by knowledge workers – an acceptance study. In L. Uden, I.-H. Ting, & J. M. Corchado (Eds.), *Knowledge management in organizations* (pp. 445-454). Springer International Publishing.
- Vincent, P., Iijima, K., Driver, M., Wong, J., & Natis, Y. (2019). *Magic quadrant for enterprise low-code application platforms*. Gartner, Inc.
- Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376 - 381.
- Wong, J., Driver, M., & Vincent, P. (2019). *Low-code development technologies evaluation guide*. <https://www.gartner.com/en/documents/3902331>. Gartner, Inc.