International Journal of Engineering Education Vol. 39, No. 6, pp. 1478–1488, 2023
Printed in Great Britain

0949-149X/91 $3.00+0.00

Investigating Factors Influencing Spreadsheets Competency of Undergraduates: Domain-Based vs Tool-Centered Learning Approach*

LENA DJORDJEVIĆ MILUTINOVIĆ[1], LAZAR RAKOVIĆ[2], RADUL MILUTINOVIĆ[1] and MARTON SAKAL[2]
[1] University of Belgrade, Faculty of Organizational Sciences, Belgrade, Serbia. E-mail: lena.djordjevic.milutinovic@fon.bg.ac.rs, radul.milutinovic@fon.bg.ac.rs
[2] University of Novi Sad, Faculty of Economics in Subotica, Subotica, Serbia. E-mail: lazar.rakovic@ef.uns.ac.rs, marton.sakal@ef.uns.ac.rs

Spreadsheets have become an indispensable part of the business world, and it is crucial that they are integrated into the corresponding study programs' curricula. However, despite their widespread use in organizations and higher education courses, there are still some challenges associated with their utilization. The risk of using spreadsheets can be singled out as very significant, and it is directly conditioned by the quality of the spreadsheet model. Although these are well-known issues, their interconnection with learning-approach has not been sufficiently explored. This paper presents the research on the interdependence of the risk of use and the quality of the spreadsheet model on the one hand, and the learning approach (tool-centered vs domain-based), on the other hand. The participants in the study were from two universities, fourth-year students of Business Informatics and Operations Management students. Both groups of students have taken courses that involve working with spreadsheets. The spreadsheet course for Business Informatics students was based on a tool-centered learning approach, while Operations Management students undertake more domain-based learning approach. Furthermore, Business Informatics students took the spreadsheet course after several programming and database courses, and therefore had a solid IT background. Operation management students took multiple courses related to problem-solving, engineering, and quantitative methods. The total number of participants was 60. Contrary to the assumption that students with greater IT background would create higher-quality spreadsheets, the results indicate no significant difference in quality between students with and without an IT background. Additionally, domain-based learning did not result in better-quality spreadsheet models. The study also explored students' attitudes toward spreadsheet risks. Students who were more familiar with spreadsheet risks, best practices and model development frameworks, created higher-quality spreadsheets with fewer errors. Consequently, incorporating lessons related to spreadsheet risks and best practices in spreadsheet courses can positively impact the quality of students' spreadsheet models. Finally, the study highlights the need for end-users and organizations to be aware of the risks that spreadsheet applications can pose in business decision-making. This paper should point out the importance of necessary improvements in curriculum of spreadsheet courses since it is evident that the mistakes have been made in training spreadsheet end-user developer for more than 30 years.

Keywords: spreadsheets courses; spreadsheet models; spreadsheet errors; higher education institutions

1. Introduction

Continuous organizational changes often lead to a mismatch between the functionality of information systems and real business needs. As a result, employees, especially knowledge workers, are often forced to use different workarounds to do their job. These workarounds are mostly not approved and supported by the official IT department and represent a grey area in IT that is often referred in the literature as Shadow IT [1], business-managed IT [2], or lightweight IT [3]. Corporate BYOD (Bring Your Own Device Solution) policies that allow employees to bring and use their personal devices for work [4] should not be confused with Shadow IT, which mostly represents the use of certain tools that are off the IT sector radar [5]. Klotz, Westner, and Strahringer [6] point out that the growing need for employee agility and customized solutions has led to the increasing prevalence of Shadow IT. The implementation of ERP solutions should eradicate this phenomenon, but in practice, it is often the opposite because these solutions become breeding grounds for Shadow IT [7, 8] and often coexist with formal systems [9]. Mallmann & Maçada [10] emphasize that the use of Shadow IT is increasingly growing. Risks associated with Shadow IT mainly include information security risks, information leakage or loss, and privacy [4]. Despite numerous challenges and risks, Shadow IT represents a source of numerous opportunities [11, 12] in terms of improving employee performance and innovation [11, 13–16].

One of the most common representatives of Shadow IT are spreadsheet applications [1], which users often create without the approval and support

* Accepted 8 August 2023.

of IT sector. Spreadsheet programs, with the most commonly used MS Excel, represent a low-code, no-code environment in which it is relatively easy to create various applications [17] and it is a part of digital workplace [18–19]. Considering the widespread use of spreadsheet applications, the familiarity of end-users with them, and the fact that spreadsheets represent one of the most convenient tools for structuring and presenting data [20], it is not surprising that the spreadsheet environment is an essential part of the daily activities of most knowledge workers. This applies to the engineering as well, where spreadsheets are used as a "powerful platform for performing engineering calculations" [21].

Consequently, for approximately three decades universities have included spreadsheet applications in different subject curriculums, since this is what companies use and expect from employees. Higher Education Institutions (HEIs) should always be aware of current and future employers' demands and ensure that programs meet the expectations of employers, especially for recent graduates. The study by Rebman et al. [22] examined the requirements for spreadsheet application familiarity in job listings, related to employees with two or fewer years of work experience, between 2019 and 2021. The results showed that Microsoft Excel is still the most required spreadsheet application by employers and that faculties should not change MS Excel training, even though there are many spreadsheet alternatives available today (like Google Sheets, LibreOffice Calc, Apple Numbers, as well as Microsoft Power Apps). Spreadsheet applications are evolving, and various companies, including Microsoft, are attempting to provide more capabilities to users who develop spreadsheets, for example with QlikView, Tableau, and PowerBI. Although these tools currently provide more power to end-users, their primary focus is on empowering users to create new analyses rather than assisting them in maintaining those analyses. As a result, Hermans et al. [23] predict that this new breed of end-user programming tools will once again experience maintainability challenges and will therefore require testing and caution.

Spreadsheets are commonly perceived as user-friendly applications suitable for self-taught end-users, but this belief is inaccurate and requires reconsideration, especially in educational environments [24]. The study conducted by Csernoch et. al. [24] addresses whether students in HEI require "simple" instructions for spreadsheet programming and whether knowledge gained through "serious" programming courses could be applicable in spreadsheet environments. Two questions can be raised here: whether spreadsheet programming is really "simple" and whether it is acceptable in today's world, where end-user programming is a ubiquitous phenomenon, to teach only IT students about software application and model development methodologies. Although this topic is relevant today, it was already discussed three decades ago. Soloway (1993) spoke in the early '90s about the need for end-users to learn programming, stating that programming is ubiquitous, and should be expanded to end-user computing. The same author proposes the idea of creating a computational medium that makes programming easier to learn and do and requires expressiveness and usefulness. Although this idea has not been implemented or popularized to the extent it should be, spreadsheets have largely taken on the role of a computational medium tool for end-users from different areas of expertise. Consistent with this, Grossman et al. [25] confirmed: "There is clear potential to generate significant benefits by developing improved methodologies for many of the very important activities performed by millions of people who interact with spreadsheet information systems.". Several authors have attempted to contribute to this topic and problem [26–30], but none of these have really taken off, neither in practice nor in education. Some spreadsheet standards do exist (http://www.fast-standard.org/, http://www.ssrb.org/standards, http://www.spreadsheetsafe.com/) [31], but they have not yet been widely adopted [23]. Moreover, students are generally not even aware of these standards, and thus cannot disseminate them when they become employed in companies.

Spreadsheet errors and the risk associated with spreadsheet programming can be highlighted as one of the most significant topics when it comes to spreadsheets [23, 32, 33]. This topic may be even more significant and interesting for practice because it results in financial losses. All the aforementioned problems related to spreadsheets have been known and discussed for about 30 years, but we have not made much progress in solving them, or the solutions have not taken root.

Spreadsheets are the tool of choice for many individuals when it comes to manipulating data, as they serve as widely used applications for end-users, enabling them to perform actions that can be considered a form of non-traditional programming within the context of spreadsheets. Those who use spreadsheets to program are often considered end-user programmers, who typically lack formal training in programming but still engage in programming to some degree. It is estimated that there are approximately 11 million end-user programmers in the United States alone, compared to only 2.75 million professional programmers [34, 35]. These figures have likely grown since the initial estimate was made [24].

The widespread use of spreadsheet programs within organizations and educational institutions unfortunately is not enough to contribute to solving problems related to spreadsheets, although they are one of the most common forms of shadow IT and end-user tools. It is even more disturbing that spreadsheets often contain an unacceptably large number of errors [36]. Reviewing the literature on spreadsheet risks and errors, it can be observed that they persist despite the wide range of specialized spreadsheet courses available [36–43]. Additionally, despite numerous proposals of good practices in spreadsheet development and use [44–49], spreadsheet errors and risks continue. One reason is certainly the overconfidence of end-users, which can be mitigated by raising awareness of spreadsheet risks [41, 50, 51]. There is no doubt that spreadsheet courses, which exist within educational institutions and beyond, are a starting point for combating spreadsheet risks. However, an analysis of available spreadsheet courses shows that they mostly focus on technical aspects of spreadsheet programs, with little or no emphasis on best practices and spreadsheet risks. Do over three decades of spreadsheet risk existence not indicate that we are still not doing something right? This is precisely what prompted the authors to conduct research.

Shadow IT is all around us, and it is an important topic in the era of digitalization. One aspect of the Shadow IT problem can be solved by reducing the number of errors in spreadsheets, both qualitative and quantitative. It is not just about formulas and functions, but also in the way, people approach working with spreadsheets, without rules and procedures. End-user programming has a dominant number of programmers compared to professional programmers. However, education in higher education institutions does not support this and does not produce new employees who can solve these problems. Correspondingly, this study aims to investigate the following hypotheses:

- Students' attitudes regarding the influence of spreadsheet errors on decision-making will be improved if they are familiar with spreadsheet risks and spreadsheet best practices.
- The quality of spreadsheet models developed by students majoring in computer science will be of a high level, with a low potential for spreadsheet errors.
- Students who are exposed to domain-based spreadsheet learning will develop spreadsheet models of higher quality than students who undergo a tool-centered approach.
- After introducing students to spreadsheet risks, related taxonomies, and best spreadsheet practices guidelines, they will develop higher-quality spreadsheet models regardless of the applied learning approach.

2. Related Research

The topic of spreadsheets in education has been discussed in numerous papers from various fields. For example, a search on WoS (Web of Science) for the keywords "spreadsheet education" yields 774 results, with the oldest paper dating back to 1996. The fields associated with these papers are diverse, ranging from Management; Operations Research & Management Science; Computer Science, Information Systems; Information Science & Library Science; Software Engineering, Business, Finance; over Engineering, Mechanical; Electrical & Electronic; Regional & Urban Planning; Urban Studies; Ecology; up to Nursing; Immunology; Infectious Diseases; Virology; Obstetrics & Gynecology; Reproductive Biology, Radiology, Nuclear Medicine & Medical Imaging; Veterinary Sciences and many others. A search for just the word "spreadsheet" yields 10,940 results in this database. A search for the keywords "spreadsheet error" yields 923 hits in this database, while "spreadsheet error education" yields only 23 results. When it comes to Scopus database the search of the results of the same keyword in a number of hits is presented in Table 1.

In accordance with the numbers presented in Table 1 and the discussion in the introduction, it is evident that spreadsheets are ubiquitous in the business world and extremely important. It is also clear that they are heavily utilized in education, and it has long been known that the way spreadsheets are taught in universities is inadequate. Scientists have attempted to draw attention to the frequency and danger of spreadsheet errors through various methodologies, tools, and approaches to learning and working with spreadsheets. However, these efforts have not produced satisfactory results in practice, nor have they been fully recognized or implemented. Therefore, it is necessary to continue research on this topic and attempt to contribute to resolving the problem.

According to Frownfelter-Lohrke [32], the design of spreadsheets often contains errors. Research conducted in the United States suggests that these errors arise due to a failure to implement

Table 1. Search results of Scopus database

Keyword	Hits Number
Spreadsheet education	1,751
Spreadsheet	21,188
Spreadsheet error	1,621
Spreadsheet error education	78

good spreadsheet design principles, which can lead to the emergence of errors. The errors in spreadsheets can be attributed to several factors, such as not separating data and formula cells, entering fixed values into formulas, failing to identify variables and name them, not paying attention to the use of absolute and relative addresses, omitting rows and columns in data groups/calculations for aesthetic purposes at the expense of the functionality of the spreadsheet, and failing to provide documentation related to how spreadsheets work [32].

One of the spreadsheet learning approaches, proposed in recent papers, is computational thinking. Muchsini et al. [52] suggest that due to the significant impact of spreadsheet errors, designing spreadsheets requires skills that support the use of technology, particularly spreadsheets and computational thinking. The authors found that many college students lack familiarity with the principles of good spreadsheet design, increasing the risk of errors in their spreadsheet designs. Therefore, the authors suggest that interventions promoting the implementation of computational thinking in the learning of spreadsheets for college students are necessary to teach good spreadsheet design. Tabesh [53] suggests that computational thinking is an essential skill set that is required throughout the application development process. This includes problem formulation skills and the ability to solve problems effectively. Accordingly, computational thinking is also necessary for designing spreadsheets, such as determining the desired outputs, identifying necessary input data, integrating relevant data and formulas into the appropriate worksheets, and assessing the efficiency of spreadsheet design automation. Instructors should provide support to students to develop computational thinking skills when using computers for data processing, editing, and presentation tasks within the system or application development process [54].

According to a study conducted by Csernoch et al. [24], it is important to recognize the value of developing computational thinking skills and to address common misconceptions of self-taught end-users and spreadsheets as user-friendly applications, particularly their application in educational environments. Building effective computer problem-solving skills and the ability to transfer knowledge is a time-consuming process that requires consciously developed and effective methods, as well as teachers who acknowledge the incremental nature of science. The authors argue that traditional tool-based approaches do not lead to the development of long-lasting knowledge or problem-solving abilities of students. Nonetheless, the authors conclude that teaching spreadsheet programming, as with other programming languages, requires direct and substantial instructional guidance.

According to several studies [55–60], teaching spreadsheeting from a programming perspective is more effective than the commonly used tool-based approaches that focus on low mathematical ability. The surface-level methods that concentrate on teaching the user interface, typing in spreadsheet tables, using wizards and help features, providing coursebooks with a large number of functions, and offering software with new features do not support the construction of schema. Without schema built up in long-term memory, both fast and slow thinking cannot be effectively applied [61], which can result in erroneous spreadsheet documents [62], ultimately leading to significant financial losses in terms of human and machine resources [63].

The research by Nagy et al. [64] indicates that students have a high level of confidence in their spreadsheet knowledge, particularly among those who have studied them in school. However, most students scored pre-structural levels on the test, which indicates a lack of recognizable knowledge. The study suggests that the decontextualized, tool-centered, low-mathability methods used in computer science education are insufficient for developing students' problem-solving and computational thinking skills [55–59, 65–66]. Students base their self-assessment on the number of different activities completed in class, with typing spreadsheet tables being the most common. It appears that studying spreadsheets is solely for learning the user interface without any meaningful goals. Based on the test results, it can be concluded that spreadsheet education does not promote the development of students' computational thinking and algorithmic skills at this stage.

The researchers not only studied spreadsheet errors but also addressed the issue of spreadsheet developers' excessive self-assurance. One of the ways to reduce the number of errors is to make developers aware of their existence. Studies by Benham and Giullian [67], Panko [41], and Raković [68] all demonstrate that spreadsheet developers tend to be overconfident and underestimate the number of errors in their work. However, Panko's study found that warning developers about errors can reduce their overconfidence. Purser and Chadwick [69] discovered that the ability to identify errors is linked to spreadsheet experience and knowledge of specific types of errors. Based on the research of Bewig [70], Raković [47, 68], and Kulesz and Zitzelsberger [71], it has been suggested that raising the awareness of spreadsheet developers about spreadsheet errors and risks and encouraging them to follow specific design guidelines could help reduce errors and improve the overall quality of spreadsheets. Kulesz and Zitzelsberger [71] also highlight that implementing best practices and guidelines can be especially effective in improving spreadsheet quality.

The above-described studies and conclusions of individual authors served as the starting point for problem definition and research design. The authors of this paper attempted to integrate the assumptions described in recent papers by proposing the hypotheses within the introduction section and testing them on two groups of students who can be considered representative based on the faculties they study and the subjects they have taken.

3. Research Methodology (Sample and Design)

To examine the set research hypotheses a quasi-experiment was conducted during the year 2022. The participants were fourth-year students of Business Informatics undergraduate studies at the Faculty of Economics (EF) of the University of Novi Sad, and Operations Management students from the Faculty of Organizational Sciences (FOS), University of Belgrade. Both groups of students have taken courses that involve working with spreadsheets.

Being a "bridge" between business/economics and IT/computer science, business informatics falls under the realm of technical and applied sciences, which is reflected in the subjects that students of business informatics take. Many of them belong to the domain of engineering, providing the multidisciplinary nature of business informatics, which encompasses knowledge and skills in computer science, as well as relevant scientific and professional topics from the fields of management and business/economics. In addition to learning several programming languages and databases, students also learn about software development processes, software analysis and design, and software testing, which prepare them to develop high-quality business/enterprise software products. These students have an IT background and were subjected to a tool-based approach for learning spreadsheets.

The module of Operations Management (OM) is part of the Management and Organization program, at the University of Belgrade, Faculty of Organizational Sciences. It is accredited within the technical and technological sciences. The expected outcomes of the module include gaining competitive knowledge, abilities, and problem-solving skills related to managing production and service operations and logistics processes at the enterprise level and within the entire supply chain. This is supported by the application of an engineering approach to management and the concept of resource efficiency, as well as the use of quantitative methods and techniques, information and communication tools, and digital technologies. Some of the subjects that OM students have taken include basic programming, optimization methods, operational research, control systems, enterprise information systems, etc. The students have learned about problem-solving, engineering, and quantitative approaches through multiple courses, which qualifies them as domain-based learners for spreadsheet modeling.

The research was divided into two iterations. In the first iteration, students were given a domain-free task to create a simple spreadsheet model. The time required to complete the assignment was limited to 20 minutes. After completing the model, students filled out a questionnaire with specific statements regarding model development, ratings of experience working with spreadsheet programs, and opinions on whether errors in Excel workbooks could affect decision-making in the organization. Then, a lecture on spreadsheet best practices and spreadsheet risks was given to the students. In the second iteration, students repeated the same task and filled out the questionnaire again. After the research was conducted, a rating of the spreadsheet model was completed for all students. The total number of participants in the research was 60. The structure by institution and gender is given in Table 2.

Starting point for the evaluation of spreadsheet models created by students were categories proposed by Rakovic et al. [51]: accuracy of output results, errors in model setup and data arrangement, hardcoding errors, use of validation, separation of inputs and outputs, differently formatted inputs and outputs and similar formatting techniques, and use of complex formulas. Accuracy of output results and errors in model setup point to the existence of quantitative errors. While other categories refer to what are called qualitative errors [72–74], which can negatively influence the use of spreadsheets and may result in quantitative errors in the future. Additionally, the authors of this paper introduced the evaluation of model design quality through six dimensions described in the next section.

Table 2. Number of participants in the study

Institution	Gender	Total number of participants
FOS	M	3
	F	23
EF	M	18
	F	16

4. Results and Data Analysis

In terms of the spreadsheet model quality, the models were observed from six dimensions (Table 3): documentation, worksheet protection, conditional formatting, data validation, application of complex formulas, and naming worksheets. If we analyze the use of these techniques (dimensions) in the first iteration of the experiment, in four out of six dimensions there are no statistically significant differences between students of the Faculty of Organizational Sciences and the Faculty of Economics, and where there are, they are not significant. Consequently, we can conclude that the IT background of the spreadsheet developer does not have an impact on the quality of the spreadsheet models (Table 3).

Table 3 (Documentation, Worksheet protection, Conditional formatting, Data validation – iteration 1): Since the assumption that the expected frequency is 5 or more in at least 80% of the cells (in this case, 50%) was not met, instead of the value of the chi-square test, the value of the Fisher's exact probability test was used, which is (1, n = 60), p is not less than 0.05, meaning that there is no statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variables documentation, worksheet protection, and data validation.

Table 3 (Application of complex formulas – iteration 1): Since the assumption that the expected frequency is 5 or more in at least 80% of the cells (in this case, 75%) was not met, instead of the value of the chi-square test, the value of the Fisher's exact probability test was used, which is (1, n = 60) $p = 0.016$, which is $p < 0.05$, meaning that there is a statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variable "application of complex formulas" with a moderate effect size measured by the phi coefficient, which is phi = 0.33 (moderate effect size according to Cohen's (1988) criterion, greater than 0.3 and less than 0.5).

Table 3 (Naming worksheets – iteration 1): Since the assumption that the expected frequency is 5 or more in at least 80% of the cells (in this case, 50%) was not met, instead of the value of the chi-square test, the value of the Fisher's exact probability test was used, which is (1, n = 60) $p = 0.031$, which is $p < 0.05$, meaning that there is a statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variable "naming worksheets," with a moderate effect size measured by the phi coefficient, which is phi = 0.31 (moderate effect size according to Cohen's (1988) criterion, greater than 0.3 and less than 0.5).

Half of the six types of spreadsheet errors examined showed no statistically significant difference between students subjected to domain-based spreadsheet learning from FOS and those whose spreadsheet instruction was mostly tool-centered from EF. Therefore, both groups of students equally have inaccurate results in their model, make mistakes in formulas or functions, and create hardcoding errors. The advantage in terms of qualitative errors such as separating input data into a separate worksheet and poorly formatted documents is on the side of FOS students, while in terms of data arrangement, the advantage is with the models created by EF students. Based on the results presented in Table 4 it can be concluded that learning domain-based spreadsheet development does not affect the number of errors that users will make in developing spreadsheet models.

Table 4 (Incorrect result, Formula/function error, hardcoding errors – iteration 1): Since the assumption that at least 80% of cells (in this case 100%) have an expected frequency of 5 or more is fulfilled, based on the value of the chi-square test (with Yates continuity correction) which is c2 (1, n

Table 3. Spreadsheet models quality analysis

Institution	Documentation		Worksheet protection		Conditional formatting		Data validation		Application of complex formulas		Naming worksheets	
	It. 1	It. 2	It. 1	It.	It. 1	It. 2	It. 1	It. 2	It. 1	It. 2	It. 1	It. 2
EF	0.0%	32.3%	2.9%	29.4%	0.0%	5.9%	2.9%	17.6%	29.4%	70.6%	0.0%	88.2%
FOS	7.7%	30.8%	11.5%	53.8%	0.0%	19.2%	19.2%	26.9%	3.8%	19.2%	15.4%	50.0%

Table 4. Analysis of errors in spreadsheet models

Institution	Quantitative errors				Qualitative errors							
	Incorrect result		Formula/function error		Hardcoding errors		Separation of inputs and outputs		Poor data arrangement		Poorly formatted documents	
	It. 1	It. 2	It. 1	It. 2	It. 1	It. 2	It. 1	It. 2	It. 1	It. 2	It. 1 4	It. 2
EF	29.4%	17.6%	17.6%	5.9%	79.4%	11.8%	0.0%	70.6%	91.2%	20.6%	100.0%	8.8%
FOS	50.0%	7.7%	26.9%	0.0%	53.8%	53.8%	26.9%	46.1%	57.7%	53.8%	34.6%	65.4%

= 60), and the significance value p is not less than 0.05, there is no statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variables incorrect result, formula/function error and hardcoding errors.

Table 4 (Separation of inputs and outputs – iteration 1): Since the assumption that at least 80% of cells (in this case 50%) have an expected frequency of 5 or more is not fulfilled, instead of the chi-square test, the value of Fisher's exact probability test was used, which is (1, n = 60) p = 0.002, which is $p < 0.05$, meaning that there is a statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variable "separation of inputs and outputs," with a medium effect size measured by the phi coefficient which is phi = 0.42 (medium effect size according to Cohen's (1988) criterion greater than 0.3 and less than 0.5).

Table 4 (Poor data arrangement – iteration 1): Since the assumption that at least 80% of cells (in this case 100%) have an expected frequency of 5 or more is fulfilled, based on the value of the chi-square test (with Yates continuity correction) which is c2 (1, n = 60) = 7.457 and the significance value p = 0.006, which is $p < 0.05$, there is a statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variable "poor data arrangement", with a medium effect size measured by the phi coefficient which is phi = 0.39 (medium effect size according to Cohen's (1988) criterion greater than 0.3 and less than 0.5).

Table 4 (Poorly formatted documents – iteration 1): Since the assumption that at least 80% of cells (in this case 100%) have an expected frequency of 5 or more is fulfilled, based on the value of the chi-square test (with Yates continuity correction) which is c2 (1, n = 60) = 27.883 and the significance value p = 0.000, which is $p < 0.05$, there is a statistically significant difference between the examined students of the Faculty of Economics and the Faculty of Organizational Sciences regarding the variable "poorly formatted document", with a large effect size measured by the phi coefficient which is phi = 0.72 (large effect size according to Cohen's (1988) criterion greater than 0.5).

Within the second iteration of the experiment, resulting spreadsheet models are analyzed after students have been introduced to spreadsheet risks and spreadsheet best practices. The authors concluded that there is an increase in the quality of spreadsheet models and a decrease in the number of errors (both qualitative and quantitative) to a greater or lesser extent. For example, in the first iteration, EF students did not document their spreadsheet models at all, while FOS students only did so in 7.7% of cases. After the lecture, almost one-third of the students in both groups documented their spreadsheet models. Furthermore, when considering the naming of worksheets, EF students did not name their worksheets at all in the first iteration, while only 15% of FOS students did so. After being introduced to good practices, nearly four-fifths of EF students and half of FOS students named their worksheets. The results of the analysis of other spreadsheet quality dimensions are shown in Table 3 and Table 4. When it comes to spreadsheet model errors, there is also a noticeable improvement after students are introduced to spreadsheet risks and best practices. For example, half of the FOS students and almost 30% of EF students had incorrect results in their spreadsheet models after the first iteration of the experiment. In the second iteration, this percentage significantly decreased, with EF 17.6% and FOS 7.7% of incorrect spreadsheet models. Therefore, it can be concluded that introducing students to spreadsheet errors and best practices can have an impact on both the quality of spreadsheet models and the number of qualitative and quantitative errors.

As part of the study, students' attitudes toward the claim that spreadsheet errors can affect decision-making in an organization were examined. Students expressed their opinion on a scale of 1 to 6 (1 – cannot affect decision-making, 6 – errors can have an extremely negative impact on decision-making). The average score in the first iteration was 4.95, while after the lecture (the second iteration) it was 5.53. The Wilcoxon rank sum test identified a statistically significant increase in the belief that errors in spreadsheets can affect decision-making in the organization, with a z-value of –4.476 and a significance level of p = 0.000 ($p < 0.05$), and a moderate effect size of r = 0.41. The median result on the scale of the impact of beliefs that spreadsheet errors can affect decision-making in the organization increased from the first iteration (Md = 5.00) to the second iteration (Md = 6.00). Therefore, it can be concluded that both groups of students are aware of the existence of spreadsheet risks in decision-making. Still, this awareness increases after additional familiarity with spreadsheet risks and best spreadsheet development practices.

5. Discussion

The starting assumption of this study was that students with a greater IT background, i.e., students majoring in computer science, which are familiar with software development methodologies, programming, testing, and other aspects of software development, would create higher-quality spreadsheets. This assumption is in line with the attitudes and results of previous research on the topic considered in the papers [52, 75]. The quality of spreadsheet models can be considered from different perspectives. Based on the analysis of best practices and instructions for building quality spreadsheets, the spreadsheets in this study were analyzed based on the use of documentation, the use of features to protect worksheets, conditional formatting, data validation, the use of complex formulas, and naming worksheets [32, 33, 51, 76]. For the first four dimensions, there is no statistically significant difference between students with a greater IT background (EF) and other group of students (FOS). Regarding the use of complex formulas and naming worksheets, there is a statistically significant difference between these two groups, but surprisingly, the difference is in favor of the other group (students from FOS). Therefore, not only does an IT background and familiarity with software development practices and other IT topics not improve the quality of spreadsheets, but to some extent, students without an IT background create higher-quality spreadsheets. This is a signal that should not be ignored and certainly requires further investigation. As the students of Business Informatics and Operations Management study certain engineering disciplines, it is assumed that the results obtained in this study can be applied to students of engineering study programs.

There are several sources [64, 77] indicating that domain-based learning of spreadsheets reduces the number of errors, and that this method is superior to tool-centered spreadsheet education. What is alarming is that even though the spreadsheet model created by the students was relatively simple and small in scale, the level of all types of errors is significant, and in some cases, very high. For quantitative errors, such as incorrect results and formula/function errors, there is no statistically significant difference between the groups of students. Therefore, in terms of quantitative errors, there is no advantage to domain-based spreadsheet learning. Qualitative errors were observed based on hard-coding errors, separating input and output data, poor data layout, and poorly formatted documents. Regarding hard-coding errors, there is no difference between the two groups, while students who underwent non-domain-based spreadsheet learning separate input data into a separate worksheet less often and format the document worse. Students undergoing domain-based learning are less cautious about data layout (data separation) compared to their peers in the other group (tool-centered learning). Taking all the above into consideration, it can be concluded, surprisingly, that even domain-based learning does not affect the quality of spreadsheet applications because the number and types of errors are not influenced by this form of learning. Perhaps this result can be attributed to the fact that FOS students, who represent a domain-based learning approach, learn to model in spreadsheets by following the design and steps provided by their professors. Since they rely on templates and lectures led by their professors, if errors are not explicitly taught to them, they may not pay attention. However, after working extensively in spreadsheets, they may become overconfident.

After the first iteration of the experiment, students were introduced to spreadsheet risks and spreadsheet best practices. When they created their spreadsheet models again, there was a clear improvement in the quality of spreadsheet models and a decrease in the number of errors. Students, having in mind spreadsheet risks and best practices, started to document their spreadsheet models, and use worksheet protection and conditional formatting more often. Additionally, students used data validation, and complex formulas and began to name worksheets more frequently than before training. Similarly, familiarity with best practices and spreadsheet risks resulted in fewer incorrect results and fewer errors in formulas and functions. Also, students used constants in formulas/functions less often (hardcoding errors), paid more attention to data layout within the spreadsheet model, and formatted documents better.

In addition to the impact on the quality of spreadsheet models and the reduction of errors, awareness of spreadsheet risks and best practices has influenced students' awareness that spreadsheet errors can negatively affect decision-making in organizations. This is very important because end-users often have excessive confidence and a lack of awareness of the risks that spreadsheet applications can pose in business. When it comes to future professional programmers (Business Informatics students in this research), a "more casual" approach to spreadsheets likely stems from an underestimation of the capabilities of these tools. The reason for this is the significantly higher complexity of the content in programming and database-related subjects (which they have completed before the spreadsheet course) compared to what is studied in the spreadsheet-related subject.

6. Conclusions

Based on the research results, it can be concluded that IT background and domain-based learning approach will not have a significant impact on the quality level and error occurrence in spreadsheet models. Even after more than three decades, we have not found a panacea for spreadsheet risks and problems. Although there have been numerous attempts to mitigate spreadsheet risks, it seems that end-users and organizations, despite their daily exposure to such hazards, lack awareness of these dangers and the need to change their approach to spreadsheets. Probably, the path to a spreadsheet panacea starts with a change in approach within spreadsheet courses. At all levels of spreadsheet courses, it would be necessary to include a section related to spreadsheet risks, spreadsheet errors, and the best practices in spreadsheet development. Within the spreadsheet risks section, it is necessary to "scare" end-users, using examples from the "horror stories" section of the EUSPRIG group. In addition to getting familiar with horror stories, course participants may also be asked to search for similar situations and present them within their group. This will affect their confidence in spreadsheet models/applications and thus positively impact the quality of future spreadsheet models they will create. Furthermore, courses should contain a module that deals with spreadsheet errors. Students need to be shown the results of research on identified spreadsheet errors both in the laboratory and "real-life" spreadsheet models. In addition to research results, this module should also include the most significant taxonomies of spreadsheet errors and explanations of individual types of errors. Then, when students are familiar with spreadsheet risks and spreadsheet errors, it is necessary to introduce them to the most common problems in end-user application development. Finally, students need to be shown multiple frameworks and best practices for spreadsheet development.

Acknowledgement – The research was funded by the University of Belgrade – Faculty of Organizational Sciences.

References

1. S. Haag and A. Eckhardt, Shadow IT, *Business & Information Systems Engineering*, **59**(6), pp. 469–473, 2017.
2. A. Kopper, M. Westner and S. Strahringer, From Shadow IT to Business-managed IT: a qualitative comparative analysis to determine configurations for successful management of IT by business entities, *Information Systems and e-Business Management*, **18**(2), pp. 209–257, 2020.
3. M.-E. Godefroid, R. Plattfaut and B. Niehaves, IT Outside of the IT Department: Reviewing Lightweight IT in Times of Shadow IT and IT Consumerization, in *Innovation Through Information Systems*, F. Ahlemann, R. Schütte, and S. Stieglitz, Eds., in Lecture Notes in Information Systems and Organisation, 48. Cham: Springer International Publishing, pp. 554–571, 2021.
4. G. L. Mallmann, A. De Vargas Pinto and A. C. G. Maçada, Shedding Light on Shadow IT: Definition, Related Concepts, and Consequences, in *Information Systems for Industry 4.0*, I. Ramos, R. Quaresma, P. Silva, and T. Oliveira, Eds., in Lecture Notes in Information Systems and Organisation, 31. Cham: Springer International Publishing, pp. 63–79, 2019.
5. L. Šedivcová and M. Potančok, Shadow IT Management Concept for Public Sector, in *Research and Practical Issues of Enterprise Information Systems*, P. Doucek, J. Basl, A. M. Tjoa, M. Raffai, A. Pavlicek, and K. Detter, Eds., in Lecture Notes in Business Information Processing, 375. Cham: Springer International Publishing, pp. 65–73, 2019.
6. S. Klotz, A. Kopper, M. Westner and S. Strahringer, Causing factors, outcomes, and governance of Shadow IT and business-managed IT: a systematic literature review, *International Journal of Information Systems and Project Management*, **7**(1), pp. 15–43, 2022.
7. N. King and B. Azad, Feral Systems as Institutional Phenomena, in *Feral Information Systems Development: Managerial Implications*, D. V. Kerr, K. Burgess, and L. Houghton, Eds., in Advances in Business Information Systems and Analytics. IGI Global, 2014.
8. L. Rakovic, T. A. Duc and V. Vukovic, Shadow IT and ERP: Multiple Case Study in German and Serbian Companies, *JEEMS*, **25**(4), pp. 730–752, 2020.
9. M. Huber, S. Zimmermann, C. Rentrop and C. Felden, Conceptualizing Shadow IT Integration Drawbacks from a Systemic Viewpoint, *Systems*, **6**(4), p. 42, 2018.
10. G. L. Mallmann and A. C. G. Maçada, The mediating role of social presence in the relationship between shadow IT usage and individual performance: a social presence theory perspective, *Behaviour & Information Technology*, **40**(4), pp. 427–441, 2021.
11. D. Fürstenau, H. Rothe and M. Sandner, Leaving the Shadow: A Configurational Approach to Explain Post-identification Outcomes of Shadow IT Systems, *Business & Information Systems Engineering*, **63**(2), pp. 97–111, 2021.
12. S. Haag, A. Eckhardt and A. Schwarz, The Acceptance of Justifications among Shadow IT Users and Nonusers – An Empirical Analysis, *Information & Management*, **56**(5), pp. 731–741, 2019.
13. J. Magnusson, D. Koutsikouri and T. Päivärinta, Efficiency creep and shadow innovation: enacting ambidextrous IT Governance in the public sector, *European Journal of Information Systems*, **29**(4), pp. 329–349, 2020.
14. I. Scalabrin Bianchi, A. Vaquina, R. Pereira, R. Dinis Sousa and G. A. Dávila, A Benefit Dependency Network for Shadow Information Technology Adoption, Based on Practitioners' Viewpoints, *Informatics*, **9**(4), p. 95, 2022.
15. M. Silic, Shadow It Steroids for Innovation, *SSRN Electronic Journal*, 2015.
16. M. Silic, D. Silic and G. Oblakovic, Influence of Shadow IT on Innovation in Organizations, *Complex Systems Informatics and Modeling Quarterly*, **8**, pp. 68–80, 2016.
17. M. El-Fadel, R. Bsat and M. Adada, Use of Spreadsheets in Environmental Education: an Application for Solid Waste Management, *International Journal of Engineering Education*, **20**(6), pp. 909–919, 2004.
18. L. Raković, M. Sakal and P. Matković, Digital workplace: Advantages and challenges, *Anali Ekonomskog fakulteta u Subotici*, **58**(47), pp. 65–78, 2022.
19. L. Mićić, H. Khamooshi, L. Raković and P. Matković, Defining the digital workplace: A systematic literature review, *Strategic Management*, **27**(2), pp. 29–43, 2022.
20. A. Y. Yurin, N. O. Dorodnykh and A. O. Shigarov, Semi-Automated Formalization and Representation of the Engineering Knowledge Extracted From Spreadsheet Data, *IEEE Access*, **9**, pp. 157468–157481, 2021.

21. S. A. Oke, Spreadsheet Applications in Engineering Education: A Review, *International Journal of Engineering Education*, **20**(6), pp. 893–901, 2004.
22. C. M. R. Jr, Q. E. Booker, H. Wimmer, S. Levkoff, M. McMurtrey and L. M. Powell, An Industry Survey of Analytics Spreadsheet Tools Adoption: Microsoft Excel vs Google Sheets, in *Proceedings of the EDSIG Conference*, p. 4901, 2022.
23. F. Hermans, B. Jansen, S. Roy, E. Aivaloglou, A. Swidan and D. Hoepelman, Spreadsheets are Code: An Overview of Software Engineering Approaches Applied to Spreadsheets, in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, IEEE, pp. 56–65, 2016.
24. M. Csernoch, P. Biro and J. Math, Developing Computational Thinking Skills With Algorithm-Driven Spreadsheeting, *IEEE Access*, **9**, pp. 153943–153959, 2021.
25. T. A. Grossman, V. Mehrotra and Ö. Özlük, Lessons from Mission-Critical Spreadsheets, *Communications of the Association for Information Systems*, **20**, pp. 1009–1042, 2007.
26. S. Bonfitto, Semantic Integration of Heterogeneous and Complex Spreadsheet Tables, in *Database Systems for Advanced Applications*, C. S. Jensen, E.-P. Lim, D.-N. Yang, W.-C. Lee, V. S. Tseng, V. Kalogeraki, J.-W. Huang and C.-Y. Shen, Eds., in Lecture Notes in Computer Science, 12683. Cham: Springer International Publishing, pp. 643–646, 2021.
27. H. Dong, S. Liu, Z. Fu, S. Han and D. Zhang, Semantic structure extraction for spreadsheet tables with a multi-task learning architecture, in *Workshop on Document Intelligence at NeurIPS 2019*, 2019. https://www.microsoft.com/en-us/research/uploads/prod/2019/12/TableStructure_DI19.pdf, Accessed 15 January 2023.
28. B. Farokhi, K. Dost, G. Weber, J. Sun and C. Lutteroth, The Semantic Spreadsheet, in *2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS)*, IEEE, pp. 171–176, 2020.
29. M. Hodkiewicz, J. W. Klüwer, C. Woods, T. Smoker and E. Low, An ontology for reasoning over engineering textual data stored in FMEA spreadsheet tables, *Computers in Industry*, **131**, p. 103496, 2021.
30. N. Kochev et al., Your Spreadsheets Can Be FAIR: A Tool and FAIRification Workflow for the eNanoMapper Database, *Nanomaterials*, **10**(10), p. 1908, 2020.
31. B. Bradley and D. Jones, Old Tech + New Tech = RegTech: Excel Spreadsheets and End User Computing in a Regulated World, in *The RegTech Book*, J. Barberis, D. W. Arner and R. P. Buckley, Eds., 1st ed., Wiley, 2019.
32. C. Frownfelter- Lohrke, Teaching good Excel design and skills: A three spreadsheet assignment project, *Journal of Accounting Education*, **39**, pp. 68–83, 2017.
33. K. N. Schneider, L. L. Becker and G. G. Berg, Beyond the mechanics of spreadsheets: using design instruction to address spreadsheet errors, *Accounting Education*, **26**(2), pp. 127–143, 2017.
34. J. L. Díaz-Herrera, The Education of the 21st Century Professional., *VNU Journal of Science: Policy and Management Studies*, **33**(2), pp. 175–183, 2017.
35. C. Scaffidi, M. Shaw and B. Myers, Estimating the Numbers of End Users and End User Programmers, in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, IEEE, pp. 207–214, 2005.
36. K. McDaid and A. Rust, Test-Driven Development for Spreadsheet Risk Management, *IEEE Software*, **26**(5), pp. 31–36, Sep. 2009.
37. T. Fetzer and T. Graeber, Does Contact Tracing Work? Quasi-Experimental Evidence from an Excel Error in England, *Proceedings of the National Academy of Sciences*, **118**(33), 2021.
38. M. R. Galbreth and L. J. LeBlanc, Overcoming Spreadsheet Risk in Supply Chain Modeling, *Journal of Business Logistics*, **31**(2), pp. 21–33, 2010.
39. T. Minter and C. Correia, The governance of risk arising from the use of spreadsheets in organisations, *Risk Governance and Control: Financial Markets and Institutions*, **4**(2), pp. 7–16, 2014.
40. R. Panko, Are we overconfident in our understanding of overconfidence?, in *First Workshop on Software Engineering Methods in Spreadsheets*, pp. 43–44, 2014.
41. P. Panko, Revising the Panko-Halverson Taxonomy of Spreadsheet Risks, in *2009 42nd Hawaii International Conference on System Sciences*, IEEE, pp. 1–10, 2009.
42. R. R. Panko and R. P. Halverson, Spreadsheets on trial: a survey of research on spreadsheet risks, in *Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences*, IEEE, pp. 326–335, 1996.
43. S. Thorne, A Review of Spreadsheet Error Reduction Techniques, *Communications of the Association for Information Systems*, **25**, pp. 413–424, 2009.
44. K. Baker, S. Powell, B. Lawson and L. Foster-Johnson, *Spreadsheet Engineering Research Project (SERP)*. 2006, http://faculty.tuck.dartmouth.edu/images/uploads/faculty/serp/serp_results.pdf, Accessed 15 January 2023.
45. T. A. Grossman and Ö. Özlük, A Paradigm for Spreadsheet Engineering, in *Proceedings of EuSpRIG 2004*, 2004.
46. T. Lemon and E. Ferguson, A Practical Approach to Managing Spreadsheet Risk in a Global Business, in *EuSpRIG 2010*, 2010.
47. L. Raković, A framework for managing spreadsheet-based end user applications, *International Journal of Management and Decision Making*, **18**(1), pp. 76–92, 2019.
48. B. Weber, Strategies for Addressing Spreadsheet Compliance Challenges, in *Proceedings of EuSpRIG 2006*, pp. 17–32, 2006.
49. N. Read and J. Batson, *Spreadsheet Modelling Best Practice*. 1999, http://www.eusprig.org/Smbp.pdf, Accessed 15 January 2023.
50. D. Kulesz, V. Käfer and S. Wagner, Spreadsheet guardian: An approach to protecting semantic correctness throughout the evolution of spreadsheets, *Journal of Software: Evolution and Process*, **30**(9), p. e1934, 2018.
51. L. Raković, M. Sakal and V. Vuković, Improvement of spreadsheet quality through reduction of end-user overconfidence: Case study, *Periodica Polytechnica Social and Management Sciences*, **27**(2), 2019.
52. B. Muchsini, Exploring college students' intention to implement computational thinking in spreadsheets learning, *Pegem Journal of Education and Instruction*, **12**(4), pp. 241–252, 2022.
53. Y. Tabesh, Computational Thinking: A 21st Century Skill, *Olimpiads in informatics*, **11**(2), pp. 65–70, 2017.
54. S. Kılıç, S. Gökoğlu and M. Öztürk, A Valid and Reliable Scale for Developing Programming-Oriented Computational Thinking, *Journal of Educational Computing Research*, **59**(2), pp. 257–286, 2021.
55. R. Abraham, M. Burnett and M. Erwig, Spreadsheet Programming, in *Wiley Encyclopedia of Computer Science and Engineering*, B. W. Wah, Ed., Hoboken, NJ, USA: John Wiley & Sons, Inc., p. 415, 2009.
56. G. Csapó, K. Sebestyén, M. Csernoch and K. Abari, Case study: Developing long-term knowledge with Sprego, *Education and Information Technologies*, **26**(1), pp. 965–982, 2021.

57. G. Csapó, M. Csernoch, and K. Abari, Sprego: case study on the effectiveness of teaching spreadsheet management with schema construction, *Education and Information Technologies*, **25**(3), pp. 1585–1605, 2020.
58. P. Hubwieser, Functional Modelling in Secondary Schools Using Spreadsheets, *Education and Information Technologies*, **9**(2), pp. 175–183, 2004.
59. V. L. Takács and K. Bubnó, Mathability in Business Education, *Acta Polytechnica Hungarica*, **19**(1), pp. 9–29, 2022.
60. P. Warren, Learning to Program: Spreadsheets, Scripting and HCI, in *Proceedings of the Sixth Australasian Conference on Computing Education*, pp. 327–333, 2004.
61. D. Kahneman, *Thinking, Fast and Slow*, Farrar, Straus and Giroux, 2011.
62. R. R. Panko and D. N. Port, End User Computing: The Dark Matter (and Dark Energy) of Corporate IT, in *2012 45th Hawaii International Conference on System Sciences*, IEEE, pp. 4603–4612, 2012.
63. European Spreadsheet Risk Interest Group, *EuSpRIG Horror Stories*. 2023. [Online]. Available: http://www.eusprig.org/horror-stories.htm
64. T. Nagy, M. Csernoch and P. Biró, The Comparison of Students' Self-Assessment, Gender, and Programming-Oriented Spreadsheet Skills, *Education Sciences*, **11**(10), p. 590, 2021.
65. P. Biró, M. Csernoch, K. Abari and J. Máth, Testing Algorithmic and Application Skills, *Turkish Online Journal of Educational Technology*, (Special Issue for INTE 2015), pp. 530–536, 2015.
66. M. Csernoch, P. Biró, J. Máth, and K. Abari, Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments, *Informatics in Education*, **14**(2), pp. 175–197, 2015.
67. H. Benham and M. Giullian, Reducing spreadsheet error rates, *Issues in Information Systems*, **6**(1), pp. 28–34, 2005.
68. L. Raković, The function of spreadsheet programs in end user application development, University of Novi Sad, Faculty of Economics in Subotica, 2014.
69. M. Purser and D. Chadwick, Does an awareness of differing types of spreadsheet errors aid end-users in identifying spreadsheets errors?, in *European Spreadsheet Risk Interest Group Conference*, pp. 185–204, 2006.
70. P. L. Bewig, How do you know your spreadsheet is right?, in *EuSPRIG*, 2005.
71. D. Kulesz and S. Zitzelsberger, Investigating Effects of CommonSpreadsheet Design Practices on Correctness and Maintainability, in *European Spreadsheet Risk Interest Group Conference*, 2012.
72. L. Leon, Z. H. Przasnyski and K. C. Seal, Introducing a Taxonomy for Classifying Qualitative Spreadsheet Errors, *Journal of Organizational and End User Computing*, **27**(1), pp. 33–56, 2015.
73. S. G. Powell, K. R. Baker and B. Lawson, A critical review of the literature on spreadsheet errors, *Decision Support Systems*, **46**(1), pp. 128–138, 2008.
74. Z. Przasnyski, L. Leon and K. C. Seal, Search of a Taxonomyfor Classifying Qualitative Spreadsheet Errors, in *European Spreadsheet Risk Interest Group Conference*, 2011.
75. B. Lawson, K. Baker, S. Powell and L. Fosterjohnson, A comparison of spreadsheet users with different levels of experience, *Omega*, **37**(3), pp. 579–590, 2009.
76. N. Garrett, Textbooks for Responsible Data Analysis in Excel, *Journal of Education for Business*, **90**(4), pp. 169–174, 2015.
77. T. A. Grossman, V. Mehrotra and M. Sidaoui, A Student-Centered Approach to the Business School Management Science Course, *INFORMS Transactions on Education*, **16**(2), pp. 42–53, 2016.

Lena Đorđević Milutinović, PhD is an associate professor at University of Belgrade, Faculty of the Organizational Sciences. Her research interests include operations management, production and services management, material flow management, control systems and models, spreadsheet engineering and management, enterprise information systems and business applications. So far, she has published over 100 bibliographic units, including scientific papers in journals and conference proceedings, monographs, textbooks and auxiliary literature. She has taken part in numerous scientific, research and commercial projects related to process improvement and automation, design and development of control models and applications, data-driven decision making in various business areas. https://orcid.org/0000-0002-3758-0147

Lazar Raković is an associate professor at the University of Novi Sad, Faculty of Economics in Subotica, Department of Business Informatics. He received his PhD in Business Informatics, and teaches the following courses: Structural Programming, Object-oriented Programming, Managing Digital Organizations, and Business Analysis Techniques – Spreadsheets. He is a co-author of a textbook and a workbook in programming and has authored or co-authored over 50 papers published in journals or presented at national or international conferences. Professional interests: digital transformation, managing digital enterprises, business process management, information system development, enterprise resource systems, end-user development, and spreadsheet engineering. https://orcid.org/0000-0002-1465-588X

Radul Milutinović is an assistant professor at the University of Belgrade, Faculty of Organizational Sciences, Serbia. He holds a PhD degree in the field of Information Systems and Management. So far, he has published over 50 articles in international and national monographs, journals and conference proceedings. The fields of his interest include operations management, innovation management, innovation projects, intellectual property, new product development, open innovation platforms, idea management systems, product-service systems, etc. https://orcid.org/0000-0003-1700-8191

Marton Sakal is a full professor at the University of Novi Sad, Faculty of Economics in Subotica, Department of Business Informatics. His research interests in the application of informatics in business and economics encompass web and mobile application development, user-centered business software interface design, business analysis using spreadsheets (end-user business application development), shadow business IT, and consumerization of IT in business. In addition to teaching at all levels of study, he has also worked as an external consultant on numerous commercial projects for industry. https://orcid.org/0000-0002-2014-4605