Mobile User Experience from the Lens of Project-Based Learning

Maria Spichkova(✉)

RMIT University, Melbourne, Australia
maria.spichkova@rmit.edu.au

Abstract. This paper presents an overview of mobile application projects conducted at the RMIT University as a part of the Learning and Teaching activities within Bachelor and Master programs, in collaboration with industrial partners. We discuss the lessons learned over eight years of teaching the corresponding courses and compare the results of our student project to the trends summarised in the recently published approached from other universities and countries.

Keywords: usability · user experience · mobile application · education · learning · teaching · project-based learning

1 Introduction

To develop sustainable application that fulfils users' needs, it is crucial to understand the whole range of user experiences and expectations [6,24]. The developers require to understand users' motivations and what they perceive as useful and easy to use [5,27], but this is a complex task which is not easy to master. Even when a developer follows general guidelines and best practices, e.g., the human interface guidelines specified by Apple[1] or the design guidelines specified by Android[2], the needs of target user group should be analysed in addition. One of the approaches to understand needs of particular group of users, is to use so-called "personas" presenting composite user archetypes that covers consist of users' ways of thinking, behaviour, goals, and motivations for a specific context [15,29]. The latest systematic mapping study on the use of personas in Requirements engineering was introduced in [26].

Having theoretical knowledge on the importance of these aspects isn't as effective as learning from real-life scenarios. Real-world problems serve as the catalyst and central point for student engagement, which was confirmed by numerous studies from many areas of software and system development, see for example [8,11,37]. Collaborative active learning allows students to explore how the theories learned in the class apply in real-world context [17]. This has an especially strong effect when the real-life scenarios aren't simulated, i.e. when

[1] https://developer.apple.com/design/human-interface-guidelines.
[2] https://developer.android.com/design/ui/mobile/guides/foundations/accessibility.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 461–472, 2024.
https://doi.org/10.1007/978-3-031-63992-0_30

the work is conducted with real stakeholders for the currently existing problems that require solutions as soon as possible.

In this paper, we present an overview of projects conducted at the RMIT University as a part of the Learning and Teaching activities in the School of Computing Technologies with the focus on mobile applications and mobile user experience. We introduce how it is embedded into the curriculum, as well as discuss lessons learned over eight years of teaching the corresponding courses and present a number of projects conducted by our students in collaboration with industrial partners in the area of mobile app development.

Outline: The rest of the paper is organised as follows. Section 2 presents an overview of related work. Section 3 presents a number of examples of the student projects as well as a discussion on our observations and lessons learned, comparing them with results of other studies. Finally, Sect. 4 summarises the paper.

2 Related Work

A systematic review of teaching methods in software engineering was presented in [7]. In that study, the authors summarised the most common teaching methods with the aim of a better alignment of Software Engineering (SE) courses with industry-relevant knowledge, tools, and practices.

Project-based learning offers numerous benefits to students. When the study projects are either derived from the industry or mirror genuine industrial challenges, the learning tasks naturally adopt a problem-based approach. By the same reasons, many universities nowadays transform the teaching activities to be *studio-based*, i.e., to focus on learning through hands-on activities and developing as result particular artefact, such as UI mock-ups, Web or mobile app prototypes, project management artefacts, etc. The core difference between studio-based courses and capstone courses is in the level of the study: studio-based leaning in general doesn't have any limitations on the the year of study and can be used for introductory courses as well. In contrast to this, capstone courses are the final year courses that should provide the culminating and integrative experience of a study program. Thus, capstone projects are especially appropriate to share with the students real industry experience [9,10,28,33].

Another notable benefit of project-based learning is offering students experience in team work, helping them develop skills in coordinating and sharing responsibilities within a group [21,34,38].

There are also approaches that aim to introduce students to research in computer science to retain them in academia or to prepare them to further study [16,35]. To encourage curiosity of Bachelor and Master students to the research in Software Engineering, we suggested to include research and analysis components in the projects as a bonus task. Short (one/two-week long) research projects have been sponsored by industrial partners and focused on the topics related to the project to conduct within semester. These have to be conducted after the semester end. The successful results of the proposed *Research embedded in teaching* initiative are presented in [13,14,19,20,43,44,46,48–51].

In our previous work [41,47,52], we presented how the research on eHealth, bio-engineering and autonomous systems is embedded into the study curriculum in the STEM (Science, Technology, Engineering and Mathematics) College at the RMIT University, Melbourne, Australia. We introduced a general structure of our Work Integrated Learning (WIL) modules and the core ideas on how the research and development activities is covered within the final-year courses. In our other work [45], we also discussed the core ideas of our redesign of the Software Engineering (SE) project course to meet such challenges as management of large cohorts of students, providing a real industrial experience, helping students to identify gaps in their skills and close these gaps based on the project experience, etc. In our current work, we went further to cover the research direction of mobile user experience and to compare the results of our student project with the trends summarised in the recently published approached from other universities and countries.

3 Project-Based Learning: Mobile Applications

The project-based capstone courses focus on hands-on practical experience within the related area of study, which in our case was conducted within Bachelor of Software Engineering (BSE) and Master of IT (MIT) study programs at the School of Computing Technologies. The RMIT academics provided their expertise on project management and software engineering, especially focusing on requirements engineering aspects and problem analysis, where some further domain expertise was also provided by our industrial partners.

The project teams typically have from five to six students (in some cases we also accepted a team of four or seven students). As the students already passed the courses on programming, fundamentals of software engineering, and software engineering project management, the capstone project focuses on integrating all the skills and knowledge acquired from the previous courses into a relatively large project. In that capstone course the students apply project management and software development methodologies in real-life settings, assuming that the core concepts have been already studied in the prerequisite courses on Software Engineering Project Management. The structure of the course was evolving over the last eight years, to incorporate the industry shift to Agile/Scrum development as well as to accommodate the recent needs for remote/hybrid working environment.

Generally, the project proposals shared with the students aren't limited to the projects on mobile app development: the students' preferences and interests might be very diverse, therefore the range of proposals should to cover several application areas to increase students' engagement. However, mobile app development and related projects build a large proportion of the proposed capstones.

In the rest of this section we present two examples of the projects that focused on mobile app development.

3.1 Deck Logger App

This project was conducted in Melbourne, Victoria (Australia) in collaboration with Spatial Vision. The aim of the project was to develop a prototype of a mobile app for Australian fishers. The mobile user needs identified in this project are largely different from findings presented in [25] by Kanij et al. based on a case study of fisherfolk communities in Bangladesh. These differences highlight the influence of human aspectas and national culture (see also [1–4,22,23,30,39,40]), and the impacts of low socio-economic background of users.

Some of the usability aspects critical for mobile user experience have been identified in both studies, such as simple (or even simplistic) user interface with large buttons, see Fig. 1. However, these aspects are typical for apps that should be used in environments where a user has limited time to deal with application and might have not very stable (physical) standing position, like on a boat, construction side, etc. However, in our study we haven't identified any specific requirements related to literacy, digital literacy, lack of access to smartphones, etc. For example, one of the complexities with using mobile apps identified by by Kanij et al. was that most of Bangladeshi fishermen could not read the text, which made the feature "text to voice" very important in this context. In contrast to this, Australian fishers didn't have any issues with literacy and digital literacy, have smartphones, and therefore required almost completely different set of functionalities: an application to automate reporting process.

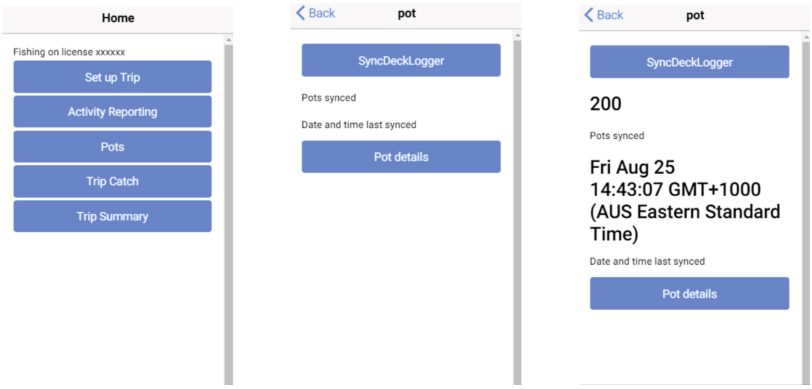

Fig. 1. Samples of the app screens for the Deck Logger App prototype: Home Screen After login, pot details page before syncing with deck logger, and pot details page after syncing with deck logger

Fisheries Victoria requires Rock Lobster fishers to report catch-and-effort data for each fishing trip which they undertake. For each trip, fishers must submit three types of reports: a pre-fishing report, a pre-landing report, and post-landing report. When these reports are manually completed by filling out

an online or paper-based logs depending on the type of report, this becomes a labour intensive and time-consuming activity for fishermen.

At the time of the project, the Victorian Rock Lobster industry applied electronic handheld devices produced by New Zealand based technology company Zebra-Tech to record catch-and-effort data. These handheld devices (so-called "Deck Loggers"[3] are easy-to-use data collection tools that require manual entry of data. However, these devices didn't offer any reporting functionality and thus manual extraction and storage of the data is required. Zebra-Tech also produces a scientific data collection device called a "Wet Tag" that can be attached various pieces of fishing equipment, e.g., a Rock Lobster Pot. These devices record location, temperature, depth and soak-time, which can be automatically transferred via Bluetooth from the Wet Tag to the Deck Logger.

The high-level objectives of this capstone project were to create a working prototype of a mobile application which:

- Enables retrieval and processing of data from existing Zebra-Tech hardware,
- Automates/streamlines the current mandatory reporting process, to decrease the time and effort fishers spend completing the reporting process,
- Enables map-based and statistical summaries of data.

different to the objectives discussed by Kanij et al. [25] in context of the case study of fisherfolk communities in Bangladesh. The reason for this was a large difference in the needs of fishers in Australia and Bangladesh, which is based on the differences in their technical and social background and industrial policies.

The app was developed using Ionic, a hybrid mobile technology that enables the application to be built for both Android and IOS using one code base. Ionic 3 was selected over other available hybrid frameworks as team members were familiar with the associated technologies and languages used within the Ionic framework. Ionic provides native user interface components for allowing for rapid prototyping and development.

3.2 Mobile Meter Reading for Non-Smart Meters

The project was conducted in collaboration with Shine Solutions Group Pty Ltd and Energy Australia, which is an electricity and gas retailing private company in Australia. It supplies electricity and natural gas to more than 2.6 million residential and business customers throughout Australia. At the time of the project, consumers used to update their utility reading manually through using an online portal. Updating the data fully manually might be inconvenient for consumers as they need to enter many details and calculate their utility reading from their meter. Also, the meters are often dusted, which makes it not so easy to read the numbers, especially for elderly people or for people with vision impairment. The cases when some digits are clipped are especially complicated.

The aim of this project was to provide a simple automated accessible alternative: to use a mobile app for capturing readings in a semi-automated way.

[3] https://www.zebra-tech.co.nz/deck-logger/.

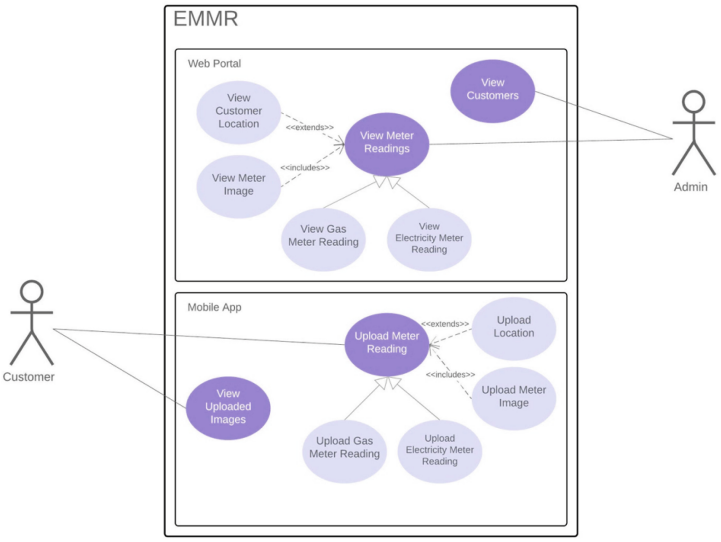

Fig. 2. Mobile Meter Reading: Use case diagram

The results of the corresponding research project, which was focused on analysis of several computer vision approaches to allow efficient automation, have been presented in [44,50]. In this paper we focus on the aspects related to the mobile user experience as well as on the corresponding learning and teaching aspects.

Figure 2 presents a use case diagram for the app. The proposed scenario was simple: a customer using the mobile application clicks an image of the meter, which will automatically start filtering out the meter readings from the image, the unrelated data will be filtered out and the best suited results are presented on the mobile app's screen. If the customer is satisfied with the image recognition results, they submit the meter reading, otherwise they might repeat an attempt. The administrator can use the web application to audit the meter readings at any time.

Figure 3 presents examples of some pages of the app. The app prototype was developed as an Android application built using React Native.

4 Discussion and Conclusions

There are many studies on project-based learning of software development. However, we believe that there are many unexplored / not discussed aspects that need to be analysed to improve learning and teaching process. In the areas like mobile user experience, where the perception of quality and the corresponding expectations are constantly evolving over recent years, the discussion of latest lessons learned is especially important.

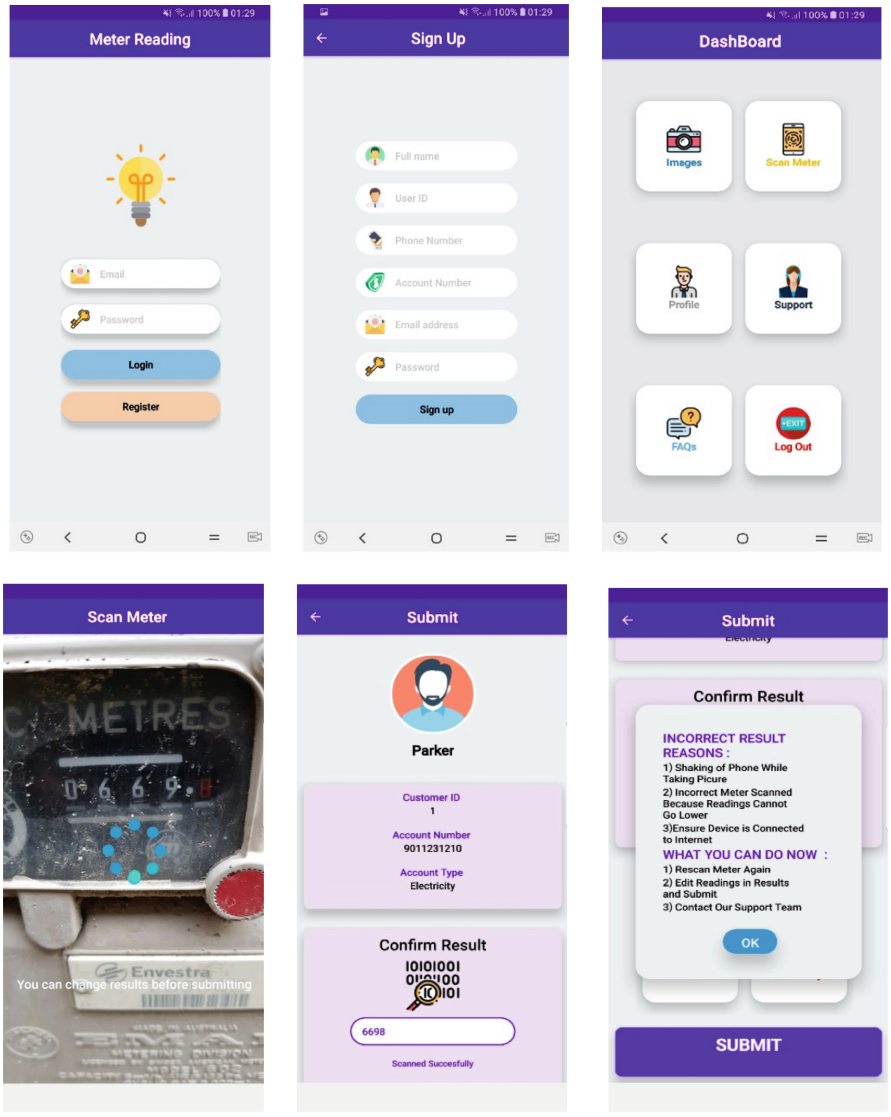

Fig. 3. Samples of the app screens: Login page, Sign Up page, User Dashboard, Capturing of the meter reading, Presenting correct results, Providing hints in the case of incorrectly identified readings

An experience report on using project-based learning in a mobile application development course was presented in [18]. The core lessons learned, identified by authors, aren't specific for mobile app development, and are rather related to general project management aspects. Similar observations have been presented in [31], where a a case study on applying user experience and user-centred design

approaches in teaching of mobile app development was introduced. We agree that these points are critical for project-based learning of software engineering (especially when the project are conducted using Agile/Scrum). Our observations also confirm that many students tend to think about software development in pure waterfall manner, even after completion of the courses on project management that have a strong focus on Agile/ Scrum. It is only once they engage in an actual project with a real industrial partner that they come to acknowledge the possibility of the system's requirements, design, and architecture really evolving throughout the course of the project.

It is generally hard for students (and potentially for for novice software engineers in general) to comprehend the fact that something might be changed over the project, as typically the assessment tasks in school and in the first year course on university level are typically well-formulated and well-structured from the beginning. Teachers aim to provide task descriptions fixed from the beginning and avoid changing them over the submission period (unless a mistake in the task is found, but this is an exceptional case). In real life this is almost never the case, as requirements should be elicited and confirmed, but then can evolve over the duration of the project. This is especially critical for the requirements related to user experience, as in many cases stakeholders might be able to provide detailed feedback only after engaging with an early prototype of the solution. Even when

- the importance of these points is highlighted in the requirements engineering literature, see e.g., [32,42], and
- the students are taught the corresponding material within the software engineering courses,

when it comes to a real project with real stakeholders and real user experience issues, many students are deeply surprised that the adjustments are needed, that the interface might need to be re-done many times, that additional functional and especially non-functional requirements are specified as top-priority in the middle or even towards the end of the project.

Thus, our observations demonstrate that the best way to master these requirements engineering and user experience aspects is to conduct a real project within a capstone course. However, it is also important to mention that completing pre-requisite courses strictly before doing the capstone is crucial, because supervisors can only mentor the students on this stage and if the students missed a large portion on background knowledge it might be to hard to prepare.

As mentioned in [12], the study projects of this type are especially effective if conducted with real industrial clients. From what we observed over the last eight years of supervision / coordination of these projects, it's also critical to provide within the course regular mentoring activities to the students on the aspects they struggle most. Thus, our observations confirmed the points discussed in [36] on enriching the students' software development project with agile coaching. Based on our lessons learned, one of the successful strategies is to enforce having an adjusted Scrum routine for meetings:

- It is important to remind students on having daily stand-ups and to update regularly their project board (Trello or Jira, depending on what the team / industrial partner decided to curate).
- If a team struggles with presentations and explanations in the Sprint review meeting with industrial partners (where the current results should be demonstrated), an additional internal (without the industrial partners) meeting with mentor might be useful, e.g. 0.5-1h before the main meeting.
- An internal Sprint retrospective meeting after the Sprint review meeting is a critical must-have part. It might be as short as 0.5h (or even shorter if the team performs really well) but having a regular catch up and regular opportunity to discuss points to improve is very effective for students to be on track and and also to share their feelings and concerns with the mentor.

In this paper, we have presented an overview of capstone projects conducted at the RMIT University as a part of the Learning and Teaching activities, with the focus on mobile user experience. We introduced two examples of the recent projects conducted in collaboration with industrial partners. We also discussed the core lessons learned from eight years of supervising and coordination the corresponding Bachelor and Master courses, as well as compared the results to the trends summarised in the recently published approached from other universities and countries.

Acknowledgement. We would like to thank Shine Solutions Group Pty Ltd and Spatial Vision for supporting the projects. We also would like to thank students who participated in the discussed projects: Callum Pearse, Baoyun Chen, Shiran Ekanayake, Lijith Win Vijay, Harish Narayanasamy, Ashish Bhardwaj, Johan van Zyl, Lavanya Krishnamurthy, Nirav Desai, and Siddharth Sachdeva.

References

1. Alsanoosy, T., Spichkova, M., Harland, J.: Cultural influence on requirements engineering activities: Australian practitioners' view. In: Siarheyeva, A., Barry, C., Lang, M., Linger, H., Schneider, C. (eds.) Information Systems Development: Information Systems Beyond 2020 (ISD2019 Proceedings) (2019)
2. Alsanoosy, T., Spichkova, M., Harland, J.: A detailed analysis of the influence of Saudi Arabia culture on the requirement engineering process. In: Damiani, E., Spanoudakis, G., Maciaszek, L.A. (eds.) ENASE 2018. CCIS, vol. 1023, pp. 240–260. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22559-9_11
3. Alsanoosy, T., Spichkova, M., Harland, J.: Does our culture influence requirements engineering activities? Vietnamese practitioners' view. In: 2020 IEEE 28th International Requirements Engineering Conference (RE), pp. 394–395. IEEE (2020)
4. Alsanoosy, T., Spichkova, M., Harland, J.: Identification of cultural influences on requirements engineering activities. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, pp. 290–291 (2020)
5. Amin, M., Rezaei, S., Abolghasemi, M.: User satisfaction with mobile websites: the impact of perceived usefulness (PU), perceived ease of use (PEOU) and trust. Nankai Bus. Rev. Int. **5**(3), 258–274 (2014)

6. Anderson, K., Burford, O., Emmerton, L.: Mobile health apps to facilitate self-care: a qualitative study of user experiences. PLoS ONE **11**(5), e0156164 (2016)
7. Aničić, K.P., Stapić, Z.: Teaching methods in software engineering: a systematic review. IEEE Softw. **39**(6), 73–79 (2022)
8. Armarego, J.: Advanced software design: a case in problem-based learning. In: Software Engineering Education and Training, pp. 44–54. CSEET, IEEE (2002)
9. Balaban, M., Sturm, A.: Software engineering lab: an essential component of a software engineering curriculum. In: Software Engineering Education and Training, pp. 21–30. ACM (2018)
10. Bastarrica, M.C., Perovich, D., Samary, M.M.: What can students get from a software engineering capstone course? In: Software Engineering Education and Training, pp. 137–145. IEEE (2017)
11. Boud, D., Feletti, G.: The challenge of problem-based learning. Routledge (2013)
12. Bruegge, B., Krusche, S., Alperowitz, L.: Software engineering project courses with industrial clients. Trans. Comput. Educ. **15**(4), 17 (2015)
13. Christianto, A., et al.: Enhancing the user experience with vertical transportation solutions. Procedia Comput. Sci. **126**, 2075–2084 (2018)
14. Clunne-Kiely, L., et al.: Modelling and implementation of humanoid robot behaviour. In: 21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, pp. 2249–2258. Elsevier Science Publishers BV (2017)
15. Cooper, A., Reimann, R., Cronin, D., Noessel, C.: About Face: The Essentials of Interaction Design. Wiley, Hoboken (2014)
16. Cooper, S., Cunningham, S.: Teaching computer science in context. ACM Inroads **1**(1), 5–8 (2010)
17. Dharmaratne, A., Fung Fung, T., Abaei, G.: Implementing a successful collaborative active learning approach in information technology discipline. In: Chang-Tik, C., Kidman, G., Tee, M.Y. (eds.) Collaborative Active Learning: Practical Activity-Based Approaches to Learning, Assessment and Feedback, pp. 237–267. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-4383-6_10
18. Francese, R., Gravino, C., Risi, M., Scanniello, G., Tortora, G.: Using project-based-learning in a mobile application development course-an experience report. J. Visual Lang. Comput. **31**, 196–205 (2015)
19. Gaikwad, P.K., Jayakumar, C.T., Tilve, E., Bohra, N., Yu, W., Spichkova, M.: Voice-activated solutions for agile retrospective sessions. Procedia Comput. Sci. **159**, 2414–2423 (2019)
20. George, L.C., Guo, Y., Stepanov, D., Peri, V.K.R., Elvitigala, R.L., Spichkova, M.: Usage visualisation for the AWS services. Procedia Comput. Sci. **176**, 3710–3717 (2020)
21. Gol, O., Nafalski, A.: Collaborative learning in engineering education. Ph.D. thesis, Unesco, Internationa Centre for Engineering Education (2007)
22. Grundy, J., Khalajzadeh, H., McIntosh, J., Kanij, T., Mueller, I.: HumaniSE: approaches to achieve more human-centric software engineering. In: Ali, R., Kaindl, H., Maciaszek, L.A. (eds.) ENASE 2020. CCIS, vol. 1375, pp. 444–468. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-70006-5_18
23. Grundy, J., Khalajzadeh, H., McIntosh, J.: Towards human-centric model-driven software engineering. In: International Conference on Evaluation of Novel Approaches to Software Engineering 2020, pp. 299–238. Scitepress (2020)
24. Hoehle, H., Venkatesh, V.: Mobile application usability. MIS Quart. **39**(2), 435–472 (2015)

25. Kanij, T., Anwar, M., Oliver, G., Hossain, M.K.: Developing software for low socio-economic end users: lessons learned from a case study of fisherfolk communities in Bangladesh. In: 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), pp. 96–107. IEEE (2023)
26. Karolita, D., McIntosh, J., Kanij, T., Grundy, J., Obie, H.O.: Use of personas in requirements engineering: a systematic mapping study. Inf. Softw. Technol. 107264 (2023)
27. Kim, Y.H., Kim, D.J., Wachter, K.: A study of mobile user engagement (MoEN): engagement motivations, perceived value, satisfaction, and continued engagement intention. Decis. Support Syst. **56**, 361–370 (2013)
28. Knudson, D., et al.: Global software engineering experience through international capstone project exchanges. In: Global Software Engineering, pp. 54–58. ACM (2018)
29. Liu, J., et al.: A curated personas and design guidelines tool for better supporting diverse end-users. In: 46th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1606–1613. IEEE (2022)
30. Liu, X., Blincoe, K., Chhetri, M.B., Grundy, J.: Human-centric software engineering - approaches, technologies, and applications. J. Syst. Softw. **204**, 111791 (2023)
31. López, M.I.C., Cervantes, A.L.E., Martínez, G.D.L.C.: Applying user experience and user-centered design software processes in undergraduate mobile application development teaching. Int. J. Cybernet. Inform. (2023)
32. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Future of Software Engineering, pp. 35–46. ACM (2000)
33. Paasivaara, M., Voda, D., Heikkila, V.T., Vanhanen, J., Lassenius, C.: How does participating in a capstone project with industrial customers affect student attitudes? In: Software Engineering Education and Training, pp. 49–57 (2018)
34. Panitz, T.: Collaborative versus cooperative learning: a comparison of the two concepts which will help us understand the underlying nature of interactive learning (1999)
35. Peckham, J., Stephenson, P., Hervé, J.Y., Hutt, R., Encarnação, M.: Increasing student retention in computer science through research programs for undergraduates. In: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, pp. 124–128 (2007)
36. Rodríguez, G., Soria, Á., Campo, M.: Measuring the impact of agile coaching on students' performance. IEEE Trans. Educ. **59**(3), 202–209 (2016)
37. Savery, J.: Overview of problem-based learning: definitions and distinctions. Essent. Readings Prob. Based Learn. Explor. Extend. Legacy Howard S. Barrows **9**, 5–15 (2015)
38. Schoor, C., Narciss, S., Körndle, H.: Regulation during cooperative and collaborative learning: a theory-based review of terms and concepts. Educ. Psychol. **50**(2), 97–119 (2015)
39. Shahin, M., et al.: Operationalizing human values in software engineering: a survey. IEEE Access **10**, 75269–75295 (2022)
40. Shams, R.A., Hussain, W., Oliver, G., Nurwidyantoro, A., Perera, H., Whittle, J.: Society-oriented applications development: investigating users' values from Bangladeshi agriculture mobile applications. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society, pp. 53–62 (2020)
41. Simic, M., Spichkova, M., Schmidt, H., Peake, I.: Enhancing learning experience by collaborative industrial projects. In: International Conference on Engineering Education and Research (ICEER), pp. 1–8 (2016)

42. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, Hoboken (1997)
43. Spichkova, M., Bartlett, J., Howard, R., Seddon, A., Zhao, X., Jiang, Y.: SMI: stack management interface. In: 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 156–159 (2018)
44. Spichkova, M., van Zyl, J.: Application of computer vision technologies for automated utility meters reading. In: International Conference on Software Technologies, p. (to appear). SciTePress (2020)
45. Spichkova, M.: Industry-oriented project-based learning of software engineering. In: 24th International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 51–61. IEEE (2019)
46. Spichkova, M., Li, B., Porter, L., Mason, L., Lyu, Y., Weng, Y.: VM2: automated security configuration and testing of virtual machine images. Procedia Comput. Sci. **176**, 3610–3617 (2020)
47. Spichkova, M., Simic, M.: Autonomous systems research embedded in teaching. In: De Pietro, G., Gallo, L., Howlett, R.J., Jain, L.C. (eds.) KES-IIMSS 2017. SIST, vol. 76, pp. 268–277. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-59480-4_27
48. Spichkova, M., Vaish, A., Highet, D.C., Irfan, I., Kesley, K., Kumar, P.D.: Gosecure: securing projects with go. In: ENASE, pp. 587–594 (2020)
49. Spichkova, M., van Zyl, J., Sachdev, S., Bhardwaj, A., Desai, N.: Comparison of computer vision approaches in application to the electricity and gas meter reading. In: Damiani, E., Spanoudakis, G., Maciaszek, L.A. (eds.) ENASE 2019. CCIS, vol. 1172, pp. 303–318. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-40223-5_15
50. Spichkova, M., van Zyl, J., Sachdev, S., Bhardwaj, A., Desai, N.: Easy mobile meter reading for non-smart meters: Comparison of AWS rekognition and google cloud vision approaches. In: Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 179–188 (2019)
51. Sun, C., et al.: Software development for autonomous and social robotics systems. In: De Pietro, G., Gallo, L., Howlett, R.J., Jain, L.C., Vlacic, L. (eds.) KES-IIMSS-18 2018. SIST, vol. 98, pp. 151–160. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-92231-7_16
52. Young, J., Spichkova, M., Simic, M.: Project-based learning within ehealth, bioengineering and biomedical engineering application areas. Procedia Comput. Sci. **192**, 4952–4961 (2021)