33rd Annual Conference of the Australasian Association for Engineering Education (AAEE 2022)
4 - 7 December 2022 – Sydney, NSW

Redesigning Software Architecture and Design Curriculum to Promote Professional Skills Among Software Engineering Students: An Experience Report

Eduardo Araujo Oliveira, Andrew Valentine
School of Computing and Information Systems, University of Melbourne
Corresponding Author Email: eduardo.oliveira@unimelb.edu.au

ABSTRACT

CONTEXT

Professional skills have become increasingly important in software engineering education; however, this is not always reflected in today's teaching curricula (Petkovic et al., 2017). The Australian Computer Society report states software engineering and ICT students are lacking essential lifelong non-technical skills necessary to create successful software systems and that higher education institutions do not sufficiently assess professional skills as learning outcomes (ACS, 2019). Besides combining and promoting professional skills in software engineering and transmitting information, teachers in software engineering must also associate theory and practice (Matthews et al., 2012).

PURPOSE OR GOAL

We redesigned the Software Design and Architecture final year compulsory subject curriculum, which is part of the Master of Software Engineering, to promote professional skills and self-regulated learning within the subject. This paper describes and evaluates our teaching initiatives.

APPROACH OR METHODOLOGY/METHODS

New teaching initiatives and subject redesign were aimed at better promoting technical and professional skills, and at supporting self-regulated learning among students. The initiatives were organised in 5 main stages (Kennedy, 2020): (i) assessment, (ii) getting the basics right, (iii) establishing our presence regularly, (iv) tutorials, (v) teaching and learning resources. We gathered information from Student Evaluation Survey (SES), and evidence from discussion board usage in 2020 and 2021 (updated subject version) to evaluate our methods.

ACTUAL OR ANTICIPATED OUTCOMES

Student Evaluation Survey results showed an increase in general satisfaction with the subject's score (contents and delivery) increasing from 3.47 and 3.7 in 2018 and 2019, respectively, to 4.13 and 4 in 2020 and 2021 (scale goes from 1 to 5. The higher the score, the better the evaluation of the subject). These were the first two times this subject has received a score equal or above 4.

CONCLUSIONS/RECOMMENDATIONS/SUMMARY

Among the lessons learned, we can highlight that planning the subject in advance and working within an inclusive space that promotes continuous communication and collaboration between the teaching team and students are the primary activities for its success. In addition, we observed that clearer assessment guidelines and continuous feedback were able to promote professional and technical skills among students.

KEYWORDS
online teaching; flipped-classroom; software engineering

Introduction

Software development is a technical activity that requires engineers to have knowledge and experience in diverse software processes, methodologies, tools and techniques, but also to perform various functions in software projects (Matturro & Fontan, 2019). Industry and academia acknowledge the importance of these technical skills and quite often emphasise them on curricula and job descriptions. However, software development processes involve tasks in various areas, such as usability, software design, programming, software maintenance, and others (Capretz & Ahmed, 2018). These tasks quite often involve extensive collaboration and face-to-face communication with various stakeholders from different domains.

A recent report from the Australian Computer Society (ACS) states successful careers in ICT depend heavily on the skills of effective and compassionate communication, ethics, Self-Regulated Learning (SRL), collaborative teamwork, or impactful stakeholder engagement, among others (ACS report, 2019). Professional skills have become increasingly important in software engineering education, however, this is not always reflected in today's teaching curricula (Petkovic et al., 2017). Besides combining and promoting professional skills in software engineering, teachers in software engineering must also associate theory and practice (Matthews, Hin and Choo, 2012).

The Master of Software Engineering hosted by the School of Computing and Information Systems at the University of Melbourne was founded in 2010 and is an entry-to-practice degree that provides students with the necessary knowledge and skills to enter the international workplace as software engineers. Graduates are skilled in software engineering principles and have the ability to apply these skills to complex, open-ended engineering tasks and problems. The degree provides sequential study in the software engineering discipline, building on a solid foundation in programming. The master program is a three-year degree (full time) and is organised in compulsory, software engineering and elective subjects.

In this context, we redesigned the Software Design and Architecture final year compulsory subject curriculum, which is part of the Master of Software Engineering, to promote professional skills and self-regulated learning within the subject. This paper describes and evaluates our teaching initiatives.

Background and Literature Review

Software engineering has been traditionally taught through lectures (Fioravanti et al., 2018). Slide presentations have been consistently used as the main resource to support explanations in the field. In addition, as explained by Fioravanti and colleagues (2018), students generally take part in projects that have little or no connection to the practice of software engineering or real-world needs. Moreover, software engineering students are often unaware of what professional skills they have attained as part of their course structure or are unaware of what additional skills they need to microcredential as part of their preparation to be 'industry ready' ie. employable (ACS report, 2019; Oliveira et al., 2021). Further, determining ways to cultivate learners' professional competences without the direct support of teachers in digital environments is a significant educational challenge.

In order for engineering students to become skilled, self-regulated and collaborative learners, they must be provided with assessments and feedback about their own professional skill development, in addition to their mastery of domain competence. Furthermore, assessment should be regarded as the collection and presentation of a range of authentic, compelling evidence that supports the judgment about where on the continuum of not just domain expertise, but also professional skills, the learner has reached at that point in time (Griffin & Robertson, 2014).

Highly self-regulated learners, for example, adopt an active approach to their learning, where they plan, monitor and adapt their strategies, and tend to perform better (Zimmerman, 2008). However, being a self-regulated learner is an effortful enterprise. Learners need to be motivated to activate their regulatory skills (Zimmerman, 2008).

Setting goals is the first step when students are being active in relation to their learning. In this phase, usually referred to as planning or forethought in self-regulated learning theory, students set the

standards they are hoping to achieve during their learning journey. The different ways to approach and interact with achievement is known as goal orientation (Valle et al., 2003). Goal orientation impacts how students monitor and regulate their cognition and behaviour during learning.

Goals can vary a lot between students within the same subjects and courses. In any case, it's important students can monitor their initial goals and adjust their learning strategies as they continue their learning journeys. To do so, they need to reflect on their learning approaches thus far.

While educational institutions have always analysed the data of their students to some extent, the multitude of new online learning tools, applications, and resources have changed the nature of the analysis which can be carried out. As this landscape is rapidly and continuously changing, the use of rubrics and peer reviews, combined with problem-based approach, can be particularly effective to generate assessment and teaching of life-long professional skills.

Evolution of the Subject

In the Software Design and Architecture subject students study software architectures in depth and the principles, techniques, and tools for creating, developing, and evaluating software architectures. The subject was designed in 2011 and had its first offer in 2012. Since its creation, the subject has 36 contact hours, comprising of two 1-hour lectures and one 1 hour workshop per week. Total commitment is around 200h.

Between 2012 and 2015, the subject assessment was organised in project work - expected to take about 36 hours (30%), and a 3-hour end-of-semester written examination (70%). Between 2016 and 2019, project component weight was increased to 40% and the 3-hour end-of-semester written examination adjusted to correspond to 60% of final grade.

During those 9 years, project work could be developed individually or in teams of 2 or 3 and involved the creation and submission of four reports totalling about 2500 words. The four stages of the project would include nominating features of students' proposed system, architectural design and implementation, re-factored design and implementation, and performance evaluation respectively. Students would choose their own projects. Lectures were delivered on-campus and in a traditional way.

The subject received evaluation scores around 3.6 between 2017 and 2019 (Table 1). Scale goes from 1 to 5. The higher the score, the better the evaluation of the subject. Students also commented and expressed their dissatisfaction with assessments, lectures, workshops design, and lack of instruction and support to conduct projects in the subject in those years.

Table 1. Student survey experience scores for SWEN90007 subject between 2017 and 2019, responding to the question "Overall, this subject has been well-taught'"

Semester	Subject Version	Number of Responses	Student Evaluation Survey*
Semester 2, 2017	Previous version	37	3.54
Semester 2, 2018	Previous version	26	3.47
Semester 2, 2019	Previous version	33	3.70

*5-point Likert scale: 1-Strongly Disagree, 5-Strongly Agree

In 2019, new subject coordinators decided to redesign the subject to address some of received feedback in previous years in the subject. New teaching initiatives and subject redesign were aimed at better promoting technical and professional skills, and at supporting SRL among students. The initiatives were organised in 5 main stages (Kennedy, 2020): (i) assessment, (ii) getting the basics right, (iii) establishing our presence regularly, (iv) tutorials, (v) teaching and learning resources. Following, we present the stages of our redesign, explaining the dynamics of each one.

Stage 1: Assessment

Students' engagement in study is often driven by assessment (Kennedy, 2020). After reading students' previous comments about assessment workload in the subject, we submitted an official request to change the subject handbook to update the subject assessment to be 100% project-based. We wanted the team-based project to follow a problem-based learning approach and to focus on real-world problems. Early in 2020, the official handbook was released and incorporated our requests.

New assessment form was organised in four submissions: (part 1) system use cases; (part 2) implementation of domain model, data mapper, unit of work, lazy load, identity field, foreign key mapping, association table mapping, embedded value, one of the inheritance patterns, authentication and authorization patterns; (part 3) addressing concurrency issues in the system; (part 4) a report reflecting on the performance the developed system.

From 2020, we gave students the chance to select one out of two available real-world projects to be designed and developed using architecture patterns covered in the subject. In 2021, for example, one of the suggested project ideas gave students the chance to implement an online vaccine booking and management application for the administration of COVID-19 vaccines. To promote problem-based learning in the subject, projects' descriptions were left quite open. Students were given two paragraphs explaining project overview and six paragraphs explaining the application domain. Working in teams of 4, students would then expand on given descriptions by suggesting use cases for their projects. Even though the same two project ideas were shared among various groups, final products were expected to be different.

Project was detailed in a 16-page document, which included information such as project overview, programming language and architecture to follow in the project, code similarity checking, use of code repositories and collaborative tools, peer review assessment, rubrics for each part/deliverable of the project, submission deadlines, FAQ and more.

Rubrics with multiple criteria and ratings from 'inadequate' to 'excellent' were created for each deliverable part to promote continuous assessment and to provide consistent feedback to teams. Moreover, we incorporated two peer reviews in the teaching semester to support the assessment of individuals. During the teaching period (12 weeks), each student now has access to 6 different feedback rounds in the subject: 4 of them (assignment rubrics) focuses on the technical aspects of project design and development, and 2 of them (peer review) focuses on the development of non-technical personal professional skills. Six assessment criteria were designed for our peer review in other to support students monitoring and self-reflecting on professional skills such as communication, teamwork, critical thinking, time management and initiative during the development of their projects. The six assessment criteria created for our peer review process focused on asking students if themselves and their peers: (i) attends group meetings, (ii) contributes meaningfully to group discussions, (iii) completes agreed tasks on time, (iv) prepares work in a quality manner, (v) demonstrates a cooperative and supportive attitude, and (vi) contributes significantly to the success of the project.

To promote engagement and make the peer review process clearer to students, received scores during both review rounds were incorporated as part of final grades. Additionally, a grading scheme was also specifically designed for peer reviews. As shown in Figure 1, students were required to write comments to peers, read all received feedback and to write a reflection about that. We believed these additional feedback could give students an opportunity to self-regulate their learning strategies aligned with their individual goals.

Code repositories for each team were created and maintained by the [subject code] teaching team under Github Organizations. This way we could guarantee students had access to similar integration tools and infrastructure to develop their projects. This approach helped us providing consistent feedback to support the development of technical software engineering skills among them. All details for the team-based project and its assessment rubrics and deadlines were available to students one week before the official start of the teaching semester on Canvas LMS.

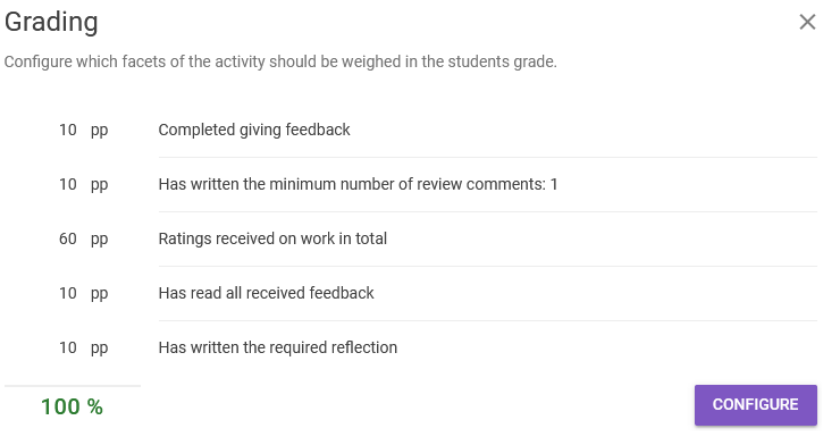

Figure 1. Peer review grading rubric

Stage 2: Getting the basics right

Before 2020, this subject page in the university LMS was used by students to watch video recordings and download slides. There was no other engagement in that space. Discussion boards were quite often empty for the entire semester.
Before redesigning this subject, student experience surveys from previous years showed students were regularly complaining about the timeliness of communication, the lack of clarity on assessment, the inability to find teaching resources on the subject site and also they didn't know much about subject planning.

In 2020, we created and adopted the use of templates in our Canvas LMS to organise our modules and pages (Figure 2). Every week, contents would be released following the same structure: lectures, discussions, assignment and additional learning resources. Specific modules were also created to provide details about projects and details about assessments.

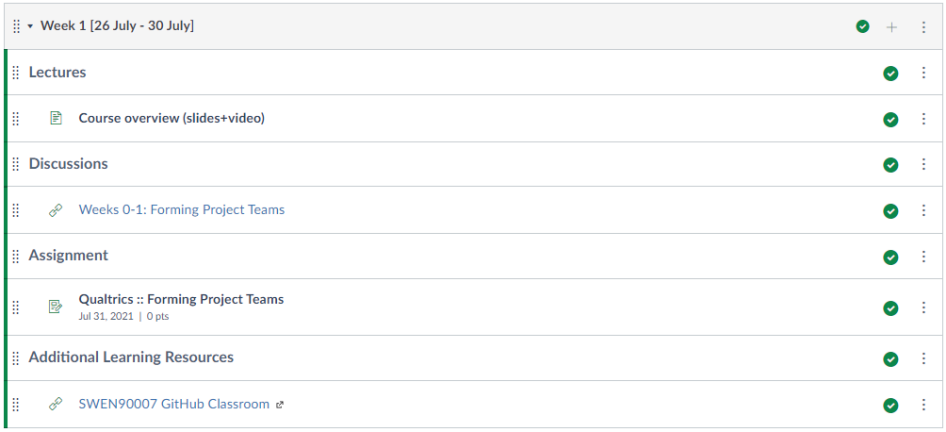

Figure 2. Canvas module template

Together with the new LMS design, a 10-minutes video was created and added to the LMS landing page of the subject to explain the new space to students. All subject information, including project details, assessments, deliverables and general expectations, was available before the start of the teaching semester.

Stage 3: Establishing our presence regularly

To create a better engaging online learning environment for our students, we (teaching team) identified the need to keep a better and more active presence in the LMS. Apart from the weekly synchronous lectures, we decided we needed to make better use of our announcements and discussion boards.

We started making use of regular announcements. At least once a week, we broadcast a message to let students know what was happening in the subject. We also adopted a strategy to reply to any new thread on the discussion board in 24h maximum (excluding weekends). We started using discussion boards for subject-based topics and to promote informal and/or general discussion (Kennedy, 2020). Teaching team started participating regularly in the discussions. We also started endorsing answers and contributions from students and acknowledging those who made valuable contributions to it.

Moreover, weekly consultation hours were highlighted in the LMS so students could make use of them. Lastly, weekly informal coffee sessions were created to promote informal catchups between students and subject coordinators.

Stage 4: Tutorials

All tutorials were redesigned to be completely aligned and focused on projects. An online book[1] was created to support students with the design and development of architecture patterns in their projects. Resources such as sample codes, tutorials and how-to documents aligned with real-world projects were created in the first semester of 2020 (prior to subject offering).

The new resources were shared among students during tutorials and also promoted during online lectures, where links were made between lecture contents and new project resources.

Stage 5: Teaching and Learning Resources

Finally, we looked across the whole subject lecture resources. Slides were updated with links to new tutorials, how-to documents, discussion board threads, and other resources available in our workshop course notes.

We also tried to promote more active learning by adopting the use of tools such as Kahoot, Polleverywhere, MIRO and IntelliJ to invite students to discuss topics and collaborate during the lectures (Lam et al., 2021). A few video recordings were released on the LMS prior to synchronous lectures between week 3 and week 11 to allow us to use these tools with students.

Evaluation of Our Teaching Initiatives

Aiming to evaluate the applied approach, we gathered information from Student Evaluation Survey (SES), and evidence from discussion board usage in 2020 and 2021. A summary of relevant information is presented below.

Overall Student Evaluation of the Subject

Our initiatives produced some interesting and promising results. For the first time, the subject obtained scores equal or above 4.0 on the SES (Table 2). In comparison, the SES scores from 2017-2019 scored between 3.47-3.70 (Table 1). This demonstrates that the subject has a consistent increase in student satisfaction after the subject was updated.

Student Feedback About our New Assessment

Several students provided written comments in the SES which provided insight into their experiences of completing the projects during the subject. Several students noted that they had benefited from industry style “project experience” (2020 SES Q9 student response 9) and “... experience in [a] real project” (2021 SES Q9 student response 2). Students also were positive towards learning the subject topics through the project based approach; “Fully project based approach really gives you hands on exposure ...” (2021 SES Q9 student response 4), “I truly loved learning the concepts through doing projects ….” (2021 SES Q9 student response 6), “The project … covers multiple aspects of the

[1] SWEN90007 Workshop Notes: https://cis-projects.github.io/swen90007_course_notes/introduction/introduction.html

software engineering methodologies, which is a good practice for the knowledge learned before." (2020 SES Q9 student response 12). These responses demonstrated that students valued being able to apply what they learnt through large scale projects. However, this was not ubiquitous. One student commented that they "... felt like the applications were too long-lived, and that perhaps it was not the best option for this subject to be a purley project-based one. … " (2021 SES Q9 student response 7).

Table 2. Student survey experience scores for SWEN90007 subject between 2020 and 2021, responding to the question "Overall, this subject has been well-taught"

Semester	Subject Version	Student Enrolments	Number of Responses	Student Evaluation Survey*
Semester 2, 2020	Updated subject	109	16	4.13
Semester 2, 2021	Updated subject	101	23	4.00

*5-point Likert scale: 1-Strongly Disagree, 5-Strongly Agree

However, there were some issues which students highlighted. Some students noted that the subject was "... a steep learning curve ..." (2021 SES Q9 student response 4) and that "the project is definitely a lot of work …" (2020 SES Q9 student response 4), suggesting that there may be additional ways to scaffold student learning at the start of the course.

Moreover, students also welcomed our peer review process. In 2020, 92 out of 109 students participated in that (completed giving feedback to peers). 64 students completed the whole grading rubric that year. This means 64 students read and reflected on received feedback. In 2021, 95 out of 101 engaged with that form of assessment, and 59 completed the whole grading rubric that year. These numbers suggest that the use of peer review was well received by most students. Although we did not investigate learning analytics data to classify students into SRL groups and correlate that information with changes in online behaviour after our interventions (peer reviews), we believe the designed peer review questions and grading process gave students new opportunities to monitor, reflect, and adjust their learning behaviour to achieve personal goals (including development of professional skills) in the subject.

Evaluation of Initiatives to Establish our Presence Regularly

Stage 3 of the subject redesign was about "establishing our presence regularly" (see above). Two sources of information were used to evaluate the better presence initiatives in the subject.

First, several students provided qualitative written comments in the SES which provided an opportunity to reflect on whether actions by the teaching team had a positive impact on students' experience. Several students provided reflections on how the teaching staff had been regularly available to meet and how this had a positive impact on their experience in the subject.

> *"... Thanks to [Subject lecturers] for making it fun and putting in A LOT of effort to create content and meet with students." (2020 SES Q9 student response 1)*
>
> *"... [Subject lecturers] have probably put the most effort into content and meeting with students than any other lecturer I've had..." (2020 SES Q9 student response 2)*
>
> *"[Subject lecturers] were very approachable and offered to have 1hr+ meetings for deliverables. Not many other lecturers are willing to do this." (2020 SES Q9 student response 6)*
>
> *"This subject provided a good way to interact with the teaching staff." (2020 SES Q9 student response)*

However, one student also highlighted that they thought lecturers could be more available right before submission deadlines.

> *"Lecturers are super helpful but wish they were responsive right before submissions" (2020 SES Q9 student response 15)*

Overall, students' written comments showed that they perceived that the teaching staff had made great efforts to be available to help answer questions, providing supporting evidence that this initiative had been successful.

Between 2017-2019 the Blackboard discussion board was not widely used, and only 15 or less posts were made per year. In contrast, during 2020, a new discussion board was fully designed and promoted on Canvas. We fragmented that discussion board to have specific threads to weekly topics covered in lectures so students could navigate through it easily. That year, Canvas analytics showed we had 304 posts organised in 27 threads. In the following year, we adopted the Ed Discussion Board and made sure we kept a similar approach to previous years. In 2021 there were 129 threads with a total of 452 posts (sum of 129 threads, 127 answers, 196 comments). This demonstrated a large adoption of the discussion board in 2020, and increased uptake during 2021. This provided further supporting evidence that the initiative to increase presence had been successful.

Challenges and Lessons Learned

Here, we present the challenges we identified to apply this approach and what we learned during our subject redesign journey. The three challenges we can highlight are:

(i) Consistent and continuous feedback: designing assessment rubrics was a very challenging task for us as we did not have much experience with that. We collaborated with educational-specialists in our university to understand how to design and incorporate these resources as part of students' learning journeys. It took us a few months and several iteration rounds to get the rubrics in a way we all believed it would provide consistent, complete, and constructive feedback to our students. Now, students are in a better position to monitor their technical and professional skills development, and to adjust their learning plans and strategies accordingly (SRL).

(ii) Assessment of individual contributions to team-based projects: managing individual marks for a team composed of 4 students was one of the greatest challenges we identified due to the high number of enrollments and projects we have in this subject. We had to come up with a scalable, consistent and evidence-based way to address this issue. We incorporated the use of peer reviews in the subject and used students' received scores to weigh the final project mark. To support peer review comments and scores, we also requested students to document team meetings (meeting minutes). We also look at individual contributions to project code repositories when necessary (in case a student receives low scores). Peer reviews and reflections generated as part of this process have been a good resource to support students developing skills such as communication, teamwork, critical-thinking and initiative.

(iii) Establishing our presence regularly: being available and establishing regular presence to large classes is not an easy task. From 2020, the teaching team had to work a lot more together and to communicate in better ways to share this workload. We have been consistently using Slack as a tool to keep regular communication in the background. Subject coordinators, lecturers and tutors share that workspace. We have fortnightly synchronous meetings to discuss about tutorials, teams and assessments, and we keep continuous (nearly daily) asynchronous conversations about conflicts in teams, questions on discussion boards and so on. When in doubt, we all discuss internally first and broadcast our agreed answer or instruction to students later.

Among the lessons learned, we can highlight that planning the subject in advance and working within an inclusive space that promotes continuous communication and collaboration between the teaching team and students are the primary activities for its success. In addition, we observed that clearer assessment guidelines and continuous feedback were able to promote professional and technical skills among students.

In future, we aim at connecting students with real industry partners to promote an even more authentic assessment in the subject and to make use of learning analytics data (LMS data) to classify students into SRL groups and correlate that information with changes in online behaviour after our interventions (peer reviews). This will give us the chance to better evaluate the impact of our provided feedback to students and its correlation with academic performance.

References

ACS Core Body of Knowledge for ICT Professionals|V1.1|22 February 2019.

Capretz, L. F., & Ahmed, F. (2018). A call to promote soft skills in software engineering. arXiv preprint arXiv:1901.01819.

de Barba, P. G., Malekian, D., Oliveira, E. A., Bailey, J., Ryan, T., & Kennedy, G. (2020). The importance and meaning of session behaviour in a MOOC. Computers & Education, 146, 103772. https://doi.org/10.1016/j.compedu.2019.103772

Fioravanti, M. L., Sena, B., Paschoal, L. N., Silva, L. R., Allian, A. P., Nakagawa, E. Y., ... & Barbosa, E. F. (2018, February). Integrating project based learning and project management for software engineering teaching: An experience report. In Proceedings of the 49th ACM technical symposium on computer science education (pp. 806-811).

Föll, P., Hauser, M., & Thiesse, F. (2018). Identifying the skills expected of IS graduates by industry: A text mining approach. In Thirty Ninth International Conference on Information Systems, San Francisco.

Griffin, P., Robertson, P. (2014). Judgement-Based Assessment. In Assessment for Teaching, pp: Cambridge University Press.

Kennedy, G. (2020). Seven ways to improve students' online learning experiences in your subject.

Hamilton, M., Carbone, A., Gonsalvez, C., & Jollands, M. (2015, January). Breakfast with ICT Employers: What do they want to see in our graduates?. In Australian Computing Education Conference (pp. 29-36).

Lam, L., Souza, R. de, Sutton, C., Oliveira, E. A., Currie, G., Hoult, R., Esfahani, L. M., Canny, L., Honig, C., & Buskes, G. (2021). Development of an online teaching-focused professional development program for junior teaching staff. Research in Engineering Education Symposium and the Australasian Association for Engineering Education Conference, 5-8 December

Matthews, R., Hin, H. S., & Choo, K. A. (2012, December). Merits and pitfalls of programming learning objects: a pilot study. In Proceedings of the 10th international conference on advances in mobile computing & multimedia (pp. 293-296).

Matturro, G., Raschetti, F., & Fontán, C. (2019). A Systematic Mapping Study on Soft Skills in Software Engineering. J. Univers. Comput. Sci., 25(1), 16-41.

Oliveira, E., Maram, V., & Sterling, L. (2021). Transitioning from motivational goal models to user stories within user-centred software design. In *RESOSY@ APSEC*.

Petkovic, D., Thompson, G., Todtenhöfer, R., Huang, S., Levine, B., Parab, S., Singh, G. & Shrestha, S. (2017). e-TAT: Online tool for teamwork and "soft skills" assessment in software engineering education. In the 40th ASEE/IEEE Frontiers in Education Conference. IEEE.

Valle, A., Cabanach, R. G., Núnez, J. C., González-Pienda, J., Rodríguez, S., & Piñeiro, I. (2003). Multiple goals, motivation and academic learning. British Journal of Educational Psychology, 73(1), 71-87. https://doi.org/10.1348/000709903762869923.

Zimmerman, B. J. (2008). Investigating self-regulation and motivation: historical background, methodological developments, and future prospects. Am. Educ. Res. J. 45, 166–183. https://doi.org/10.3102/0002831207312909.

Acknowledgements

The redesigning of Software Architecture and Design subject would not have been possible without the exceptional support of Dr Maria Read, Luke Rosa, Dr Eman Bin Khunayn and Max Plumley. Thank you very much!

Copyright statement