Bridging the Communication Gap—A Pilot Course on Interdisciplinary Requirements Engineering

Anne Catherine Gieshoff[1], Marcela Ruiz[2,*], Martin Schuler[1] and Michael Wahler[2]

[1]Zurich University of Applied Sciences, School of Applied Linguistics, Institute of Translation and Interpreting, Winterthur, Switzerland

[2]Zurich University of Applied Sciences, School of Engineering, Institute of Computer Science, Winterthur, Switzerland

Abstract

Teaching the practical aspects of requirements engineering is challenging due to the lack of effective tools, methods, and practices for teaching requirements elicitation, especially for novices. The research community has proposed various solutions to measure the effectiveness of elicitation techniques, such as interviews. However, there is limited research on teaching experiences that address both technical and communication skills.

To address this, we implemented an innovative interdisciplinary pilot course for final-year computer science and applied linguistics students at our university. Linguistics students conceptualized a language learning application, acting as the stakeholders, while computer science students elicited requirements through interviews. This collaboration provided realistic stakeholder interaction and insight into the software development process for linguistics students. This paper details the setup of the interdisciplinary pilot course, presents the course evaluation results, and shares the lessons learned from this unique educational endeavor.

Keywords

Interdisciplinary Education, Requirements Engineering, Pilot Course

1. Introduction

As part of the Bachelor's program in computer science, the Zurich University of Applied Sciences (ZHAW) offers an elective module for final year's students (CS students) named *Advanced Software Engineering* (ASE), which provides a deep dive into Requirements Engineering (RE) and software architecture during one semester. For the requirements engineering lessons, students learn to elicit requirements and document them properly by studying the syllabus of the Certified Professional for Requirements Engineering (CPRE) foundation level developed by the International Requirements Engineering Board (IREB) [1]. For the practical exercises, the participants generate an idea for a software application and use RE elicitation and specification

In: A. Hess, A. Susi, E. C. Groen, M. Ruiz, M. Abbas, F. B. Aydemir, M. Daneva, R. Guizzardi, J. Gulden, A. Herrmann, J. Horkoff, S. Kopczyńska, P. Mennig, M. Oriol Hilari, E. Paja, A. Perini, A. Rachmann, K. Schneider, L. Semini, P. Spoletini, A. Vogelsang. Joint Proceedings of REFSQ-2025 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2025. Barcelona, Spain, April 7, 2025.

*Corresponding author.

gies@zhaw.ch (A. C. Gieshoff); ruiz@zhaw.ch (M. Ruiz); shmt@zhaw.ch (M. Schuler); wahl@zhaw.ch (M. Wahler)

0000-0002-4383-190X (A. C. Gieshoff); 0000-0002-0592-1779 (M. Ruiz); 0000-0002-0857-8136 (M. Schuler); 0009-0006-5301-6315 (M. Wahler)

techniques for the case study. Whereas this proves to be more motivating than using a textbook case study, our students miss the opportunity to practice their technical and communication skills, as well as realistically confront one of the major challenges in requirements elicitation: communicating with stakeholders and getting an understanding of the problem context [2].

Pedagogical approaches such as problem-based learning, project-based learning, and challenge-based learning have been shown to be successful in helping engineering students combine knowledge acquisition, application, and development of disciplinary skills [3, 4, 5]. In the past decade, a key trend in RE education is to involve realistic non-technical stakeholders to foster communication skills when teaching RE [6]. However, replicating the use of requirements elicitation techniques in the classroom—for example, by using interview techniques—is particularly challenging, because of the lack of tools, practical exercises, and methods for effective teaching [7]. Having realistic stakeholders is cumbersome and difficult to implement as it demands a human resource intensive program and imposes conflicting objectives between students and stakeholders [8]. Simulating stakeholder by introducing role playing, prepared scenarios, or AI agents emerged as potential solutions in education [9, 10, 11]. However, teaching the development of soft skills through genuine stakeholder participation is crucial, as it helps students grasp the importance of personal communication in gathering, negotiating, and confirming requirements [12]. Thus, investigating ways to keep realistic stakeholders involved despite challenges can bring great educational benefits.

In this paper, we share our teaching experience by involving realistic non-technical stakeholders in the RE classroom by creating a common learning environment. Our pedagogical goal is to foster synergies among disciplines for better knowledge acquisition and practical implementation in a setting that is sustainable and with win-win scenarios for requirements engineering students and realistic stakeholders. To address this goal, we have implemented an innovative interdisciplinary pilot course in the Autumn Semester 2024, bringing together the CS student of the ASE course with the final year bachelor students of the *Project Plus* course at the Applied Linguistics department (L students) of our university.

This paper is structured as follows: In Section 1, we present the course description, intended learning goals, course's participants and activities. In Section 2, we present the course evaluation including the results from a pre and post survey, reflect on our observations regarding students' engagement, and discuss the challenges in teaching and implementation of the course. In Section 3, we present a summary of the main conclusions and teaching directions for involving interdisciplinary participants in the RE classroom.

2. An Interdisciplinary RE Pilot Course

Background. The bachelor program in computer science of the ZHAW School of Engineering offers a final-year elective module named Advanced Software Engineering consisting of two main parts: RE and software architecture (SA). The course's syllabus follows the materials for from the IREB Certified Professional for Requirements Engineering [1], and iSAQB Certified Professional for Software Architecture [13]. The course combines a flipped classroom approach with interactive seminars. In the classroom, the theory is reviewed under the light of case studies to practice various methods, techniques (like interviews with stakeholders), and solution approaches. Current frameworks (such as Spring Boot, Django, Angular) are discussed and used to implement the use cases.

The bachelor program in the ZHAW Applied Linguistics department offers the course *Project Plus* Software Requirements: the exciting journey from an idea to software. In this course, L students are sensitized to communication when working with other non-language-related professionals and reflect on their own role in software development projects. Students derive an idea for a language-related software application, create mock-ups that simulate the core functions, and validate this mock-up in a usability test to derive recommendations for further development.

Intended Learning Goals. The main common and specific goals for CS students and L students of the pilot course are:

1. Communicate software requirements efficiently
   CS students: Elicit software requirements and architecture from stakeholders.
   L students: Provide clear features for a language-related software application.
2. Use software elicitation tools effectively
   CS students: Apply requirements elicitation techniques to collect software requirements.
   L students: Apply communication techniques to provide software and quality features.
3. Refine software requirements and architectures
   CS students: Specify software requirements and architecture reports.
   L students: Build mock-ups and apply usability testing methods.

Teaching team. The first and third authors of this paper are the teachers of *Project Plus* course and faculty members of the Applied Linguistics department. The second and fourth authors are the teachers of the ASE course and faculty members of the School of Engineering. The conception of the joint pilot course started in the Spring semester 2024 with the objective to allow CS students and L students to realistically practice their gained knowledge and achieve the learning goals. For this, we designed both courses for parallel execution and provided joint activities with the following criteria:

- L students and CS students work together on the same project.
- Provide a setting in which stakeholders have a real stake in the project and want it to succeed.
- Stakeholders should have significant experience in a domain that is largely unknown to CS students.
- Stakeholders can invest a significant amount of time in requirements engineering.

Course participants and activities. The ASE course had 38 enrollments and the *Project Plus* had 8 enrollments. We created interdisciplinary teams of 4–5 CS students with the role of software engineers and non-communication experts and one L student as the main stakeholder and non-technical expert.

The activities of the course spanned during the Autumn semester 2024, which lasted a total of 14-weeks with four 45-minute face-to-face lessons and 5.5 hours of self-study per week. Figure 1 presents a snapshot of parallel and joint activities in the timeline.

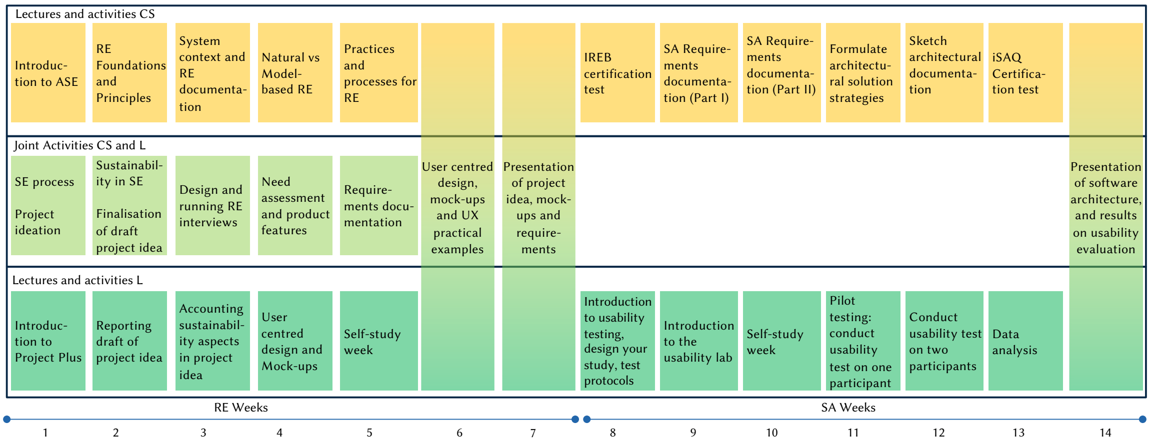


Figure 1: Main activities of the ASE and *Project Plus* courses with a timeline

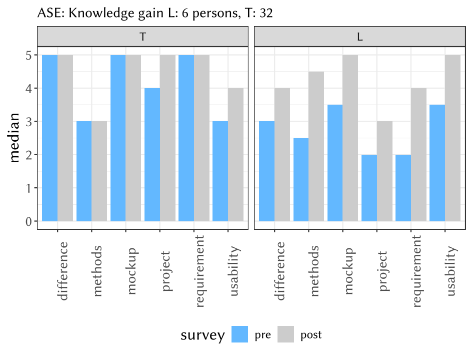
As part of the design of the course, we decided to have a strong interaction between the students during the RE weeks 2–7 (See Figure 1). CS students were provided with knowledge and tasks that allowed them to elicit, specify and describe requirements for a language-related software provided by L students. CS students were in charge to design interviews as RE technique for requirements elicitation.

This setup allowed both groups of students to benefit from each other's expertise to maximize the experience and achieve their individual but common goals: design a language-related software application. In Week 7, each team gave a presentation of the main requirements of the language-related software in terms of natural language like user stories, software models like use cases, context and conceptual models, as well as traceability matrices to show how features relate to the different requirements of the envisioned application. Additionally, L students developed the UI mock-ups. All groups accounted for sustainability aspects in the software development life cycle as proposed in [14]. During Weeks 8–13, the CS students and L students received knowledge in software architecture and usability assessment, respectively. During those weeks, there were no joint activities in the program, but the students were encouraged to keep in contact and align for a final presentation of the software architecture and usability testing results in Week 14.

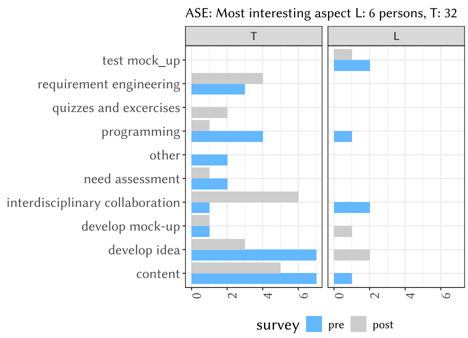
3. Course Evaluation and Reflections

Course Evaluation. We conducted two surveys at the beginning and end of the course to assess students' prior and gained knowledge, expectations, and learning goals, perception of the course's relevance, and perception of effort regarding the course's tasks and interdisciplinary group work (see results in Figure 2).

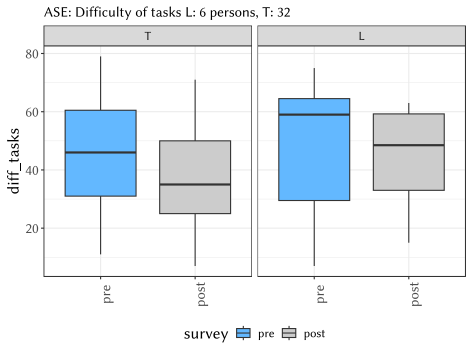
The results of the surveys showed that, in general, developing an idea and interdisciplinary collaboration was perceived as an interesting experience. The perceived knowledge gain of CS students regarding requirements engineering did not evolve compared to prior knowledge. This can be explained by the fact that we did not emphasize which aspects we expected them to



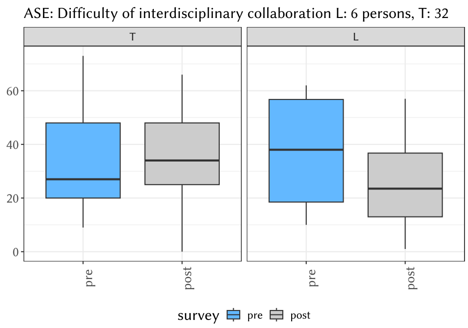
(a) Knowledge gain



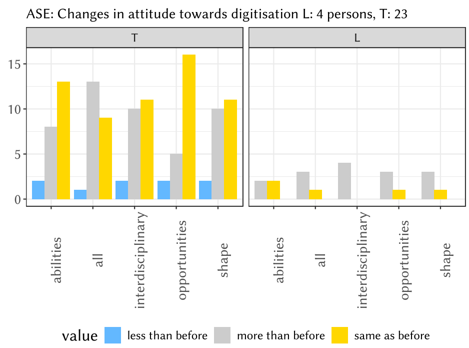
(b) Most interesting aspects



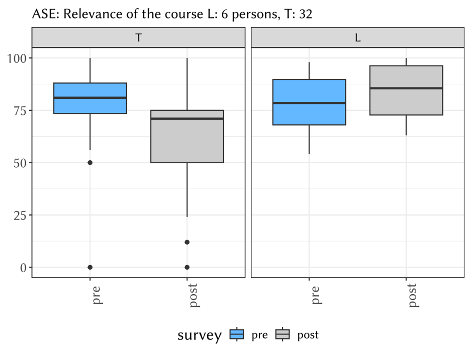
(c) task difficulty



(d) Perceived difficulty of interdisciplinary collaboration



(e) Changes in attitude towards digitization



(f) Course relevance

Figure 2: Course evaluation results

measure, such as experiencing first-hand use of requirements elicitation techniques with realistic stakeholders, specifying requirements using RE methods, etc. On the other hand, CS students perceived gains regarding software development project and usability evaluation. L students perceived knowledge gains on software requirements, methods and usability evaluation.

Student Engagement. Despite both courses not having a mandatory onsite presence during lectures and workshops, we observed that the students showed a high level of commitment and attendance compared to previous editions of the ASE course. This can be seen in the average presence of the students in the classroom, which increased from around 14 % in previous years to 47 % this year. We attribute this to the open and collaborative atmosphere that was established in the interdisciplinary setting in the first half of the semester.

The interdisciplinary background of the students allowed diverse discussions related to technical and communication aspects. The results from this experience motivate us to further evolve the syllabus and develop more tools to foster and measure the learning progress in RE.

Challenges. Finding a common time slot for courses organized by two different departments was administratively challenging. We solved this challenge by arranging a weekly teaching contact of two 45-minute slots each in which we taught CS students and L students separately. Afterwards, we brought the students together for two 45-minute joint activities. Another challenge was the differences in the grading components for the L students and CS students, which required a thorough alignment to clarify the differences between the groups despite working on the same project. The big difference in groups size imposed challenges, which we addressed by assigning one L student (in the role of stakeholder) to a team of CS students (the software engineers).

4. Summary and Next Steps

In this paper, we report on our experience in running a pilot course on interdisciplinary requirements engineering. We taught final-year bachelor's students in computer science and applied linguistics together. The course had a total duration of 14 weeks in which L students provided an idea of a language-related software application with corresponding mock-ups and usability testing. CS students were in charge to elicit requirements from the L students as realistic stakeholders, and specified requirements engineering and software architecture reports. We ran a pre-and-post survey to understand students' perceived gained knowledge and relevance. The results show that L students learned about all topics, CS students only about usability studies and software projects. The students perceived cross-disciplinary collaboration as a very interesting aspect of the course, which helped to increase engagement and onsite attendance. In general, L students perceived the course to be relevant for their career, in contrast to CS students.

For the next edition of the course, we plan to explore platforms to have a collaborative space that offers course materials and a personalized knowledge base for L students and CS students. Furthermore, we plan to review the theoretical content and exercises for CS students to position its relevance in the curriculum, as well as career opportunities. We plan to investigate strategies to foster the perceived RE knowledge for CS students, by introducing feedback tools when learning and using RE practices in the classroom. In this line, we plan to expand our investigation to review existing practices, tools, and materials provided for RE and interdiscplinary classrooms in social science courses.

References

[1] IREB, International requirements engineering board, 2025. URL: htps://ireb.org/en, date accessed: 04/02/2025.

[2] J. Coughlan, R. D. Macredie, Effective communication in requirements elicitation: A comparison of methodologies, Requirements Engineering 7 (2002) 47–60. URL: https://link.springer.com/article/10.1007/s007660200004. doi:10.1007/S007660200004/METRICS.

[3] V. Sukackė, A. O. P. de Carvalho Guerra, D. Ellinger, V. Carlos, S. Petronienė, L. Gaižiūnienė, S. Blanch, A. Marbà-Tallada, A. Brose, Towards active evidence-based learning in engineering education: A systematic literature review of pbl, pjbl, and cbl, Sustainability 2022, Vol. 14, Page 13955 14 (2022) 13955. URL: https://www.mdpi.com/2071-1050/14/21/13955/htmhttps://www.mdpi.com/2071-1050/14/21/13955. doi:10.3390/SU142113955.

[4] K. Doulougeri, J. D. Vermunt, G. Bombaerts, M. Bots, Challenge-based learning implementation in engineering education: A systematic literature review, Journal of Engineering Education 113 (2024) 1076–1106. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/jee.20588https://onlinelibrary.wiley.com/doi/abs/10.1002/jee.20588https://onlinelibrary.wiley.com/doi/10.1002/jee.20588. doi:10.1002/JEE.20588.

[5] S. E. Gallagher, T. Savage, Challenge-based learning in higher education: an exploratory literature review, Teaching in Higher Education 28 (2023) 1135–1157. URL: https://www.tandfonline.com/doi/abs/10.1080/13562517.2020.1863354. doi:10.1080/13562517.2020.1863354.

[6] M. Daun, A. M. Grubb, V. Stenkova, B. Tenbergen, A systematic literature review of requirements engineering education, Requirements Engineering 28 (2023) 145–175. URL: https://link.springer.com/article/10.1007/s00766-022-00381-9. doi:10.1007/S00766-022-00381-9/FIGURES/15.

[7] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini, B. Donati, Teaching requirements elicitation interviews: an empirical study of learning from mistakes, Requirements Engineering 24 (2019) 259–289. URL: https://link.springer.com/article/10.1007/s00766-019-00313-0. doi:10.1007/S00766-019-00313-0/FIGURES/25.

[8] G. Gabrysiak, R. Hebig, L. Pirl, H. Giese, Cooperating with a non-governmental organization to teach gathering and implementation of requirements, Software Engineering Education Conference, Proceedings (2013) 11–20. doi:10.1109/CSEET.2013.6595232.

[9] D. Zowghi, S. Paryani, Teaching requirements engineering through role playing: lessons learnt, in: Proceedings. 11th IEEE International Requirements Engineering Conference, 2003., 2003, pp. 233–241. doi:10.1109/ICRE.2003.1232754.

[10] B. Gorer, F. B. Aydemir, Generating requirements elicitation interview scripts with large language models, 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW) (2023) 44–51. doi:10.1109/REW57809.2023.00015.

[11] P. Marques, M. Silva, C. Gusmão, D. Castro, M. Schots, Requirements engineering out of the classroom: Anticipating challenges experienced in practice, in: 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), 2020, pp. 1–9. doi:10.1109/CSEET49119.2020.9206220.

[12] M. Daun, A. M. Grubb, B. Tenbergen, A survey of instructional approaches in the requirements engineering education literature, in: 2021 IEEE 29th International Requirements

Engineering Conference (RE), 2021, pp. 257–268. doi:10.1109/RE51729.2021.00030.
[13] International Software Architecture Qualification Board iSAQB, The international standard in training and certification of software architects, 2025. URL: https://www.isaqb.org/, date accessed: 04/02/2025.
[14] N. Seyff, S. Betz, D. Lammert, J. Porras, L. Duboc, I. Brooks, R. Chitchyan, C. C. Venters, B. Penzenstadler, Transforming our world through software – mapping the sustainability awareness framework to the un sustainable development goals, International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE - Proceedings (2022) 417–425. URL: https://research-information.bris.ac.uk/en/publications/transforming-our-world-through-software-mapping-the-sustainabilit. doi:10.5220/0011063200003176.