MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
ACM (Apenas Open Access)										
0	Examining Teamwork: Evaluating Individual Contributions in Collaborative Software Engineering Projects	Collaborative learning and group projects are integral to Software Engineering (SE) education as they help prepare students for professional environments where teamwork and collective problemsolving are crucial. Accurately assessing individual contributions within the group project setting remains a significant challenge. Traditional assessment methods often fail to distinguish between individual efforts and collective achievements, particularly in teambased assignments prevalent in SE courses. This paper explores how several factors potentially affect individual participation in group programming projects. We compare metrics for evaluating individual contributions and examine the effect of group formation strategies on collaborative practices. Our data consist of the collaborative artifacts of 77 teams and peer evaluation responses from 154 students across two semesters of a graduate-level software engineering course, providing insights into several research questions, including the effectiveness of different metrics for estimating individual contributions, the distribution of programming tasks among students, and the impact of group composition on task distribution and peer evaluations.	☐	☐	☐	☐	☐	☐	☐	☐
1	GameStudio Simulator: Integrating Industry Practice and Formative Individual Assessment in Collaborative Game Development Capstone Projects	Capstone courses, where students work on a large group project at the end of their degree, are common in computing. Many accreditation and industry bodies explicitly require or recommend them. Over the past two decades, capstone courses specifically focused on game development have become increasingly popular. Game development offers students the opportunity to express themselves creatively and learn to work with people from a variety of disciplines, such as programming, design, music, and art. In this article, we present the results of 10 years of iteration on a game development capstone course. This course is structured as a year-long collaboration between software engineering and game design students, but maintains separation between the disciplines in terms of supervision and assessment. Based on our experiments and observations, we provide recommendations for teaching practices, team structure, project timeline and assessment. In these, we focus on emulating and incorporating industry practices, maximising opportunities for formative feedback and maintaining authentic individual assessment. We have found that our most recent implementation has resulted in improved student satisfaction and learning outcomes compared to previous iterations.	☐	☐	☐	☐	☐	☐	☐	☐
2	Teaching Scrum with a focus on compliance assessment	TheScrumframeworkhasgainedwidespreadadoptionintheindustry for its emphasis on collaboration and continuous improvement. However, it has not reached a similar relevance in Software Engineering (SE) curricula. This work reports the experience of five editions of a SE course within an M.Sc. Degree in Computer Engineering. The course primary educational objective is to provide students with the skills to manage software development projects with Scrum. The course is based on the execution of a team project andonthedefinitionofqualitative and quantitative means of assessment of the application of Scrum. The conduction of five editions of the course allowed us to identify several lessons learned about time budgeting and team compositions in agile student projects and its evidence of the applicability of the framework to software development courses.	☑	☐	☐	☐	☐	☐	☑	☐
3	The Impact of Team Diversity in Agile Development Education	Software Engineering is mostly a male-dominated sector, where gender diversity is a key feature for improving equality of opportunities, productivity, and innovation. Other diversity aspects, including but not limited to nationality and ethnicity, are often understudied. In this work we aim to assess the impact of team diversity, focusing mainly on gender and nationality, in the context of an agile software development project-based course. We analyzed 51 teams over three academic years, measuring three di erent Diversity indexes– regarding Gender, Nationality and their co-presence– to examine how di erent aspects of diversity impact the quality of team project outcomes. Statistical analysis revealed a moderate, statistically signi cant correlation between gender diversity and project success, aligning with existing literature. Diversity in nationality showed a negative but negligible e ect on project results, indicating that promoting these aspects does not harm students' performance. Analyzing their co-presence within a team, gender and nationality combined had a negative impact, likely due to increased communication barriers and di ering cultural norms. This study underscores the importance of considering multiple diversity dimensions and their interactions in educational settings. Our ndings, overall, show that promoting diversity in teams does not negatively impact their performance and achievement of educational goals.	☑	☐	☐	☐	☐	☐	☐	☑
4	Identifying Struggling Teams in Software Engineering Courses Through Weekly Surveys	Teaming is increasingly a core aspect of professional software engineering and most undergraduate computer science curricula. At NC State University, we teach communication and project-management Computer science education is increasingly focused on team-based skills explicitly through a junior-level software engineering course. However, some students may have a dysfunctional team experience that imperils their ability to learn these skills. Identifying these teams during a team project is important so the teaching staff can intervene early and hopefully alleviate the issues. Wepropose a weekly reflection survey to help the course teaching staff proactively identify teams that may not be on track to learn the course outcomes. The questions on the survey focus on team communication and collaboration over the previous week. We evaluate our survey on two semesters of the undergraduate software engineering course by comparing teams with poor endof-project grades or peer evaluations against teams flagged on a weekly basis through the surveys. We find that the survey can identify most teams that later struggled on the project, typically by the half-way markoftheproject, and thus mayprovideinstructors with an actionable early-warning about struggling teams. Furthermore, a majority of students (64.4%) found the survey to be a helpful tool for keeping their team on track. Finally, we discuss future work for improving the survey and engaging with student teams.	☐	☐	☐	☐	☐	☐	☐	☐
5	Hints on Designing and Running Project-based Exams for a Software Engineering Course	In this paper, we share our experience in designing and running a project-based exam for a Software Engineering course at the undergraduate level. We underline the teaching objectives and content of the course which aim to prepare students for the industrial environment, e.g., various design perspectives on the project, software quality assurance and evaluation, tools for development and software quality assessment, project management and team work. We present our project-based exam and summarize hints on its design, organization, and evaluation. Project examples are also introduced.	☐	☐	☐	☐	☐	☐	☐	☐
6	Understanding Skill Transfer Between University and Workplace Through Reflective Practice: A Software Engineering Work-Based Learning Experience	Work-based learning (WBL) programs in Software Engineering provide students with opportunities to apply academic knowledge in the workplace. However, how students navigate the transition between knowledgeandskills acquired at university and their application in the workplace remains largely unexplored. We implemented a WBL Software Engineering program in which students spend most of their time in the workplace rather than at university. Reflective student essays were used to examine how our implementation supports the integration and transfer of learning between academic and professional contexts. University modules– such as those on software design patterns and programming– offered directly applicable skills, while foundational knowledge and problem-solving frameworks supported further development. Workplace experiences reinforced understanding, provided context, and enabled practical application. However, skills such as project management, leadership, and context switching were primarily developed in the workplace, revealing gaps between academic team projects and industry needs. Difficulties in transferring some advanced technical skills also point to a disconnect between academic instruction and industry application. Our findings suggest a cyclical relationship between university and workplace learning, with each enhancing the other. We propose key considerations for WBL program design, including curriculum alignment, structured skill development, and integrated delivery. Finally, we underscore the value of reflective essays in observing and supporting skill integration.	☑	☐	☐	☐	☐	☐	☐	☑
7	Preparing Autistic Students for the AI Workforce	Software project courses too often focus instruction on technical skills but leave necessary communication and teamwork skills as an exercise for the learner. When autistic students take these courses, they often find difficulty fitting into teams with non-autistic students because of their different styles of communication. To explore how to help autistic students adapt to non-autistic normative software engineering environments, we designed and taught a first-of-its-kind online project course on AI, explicitly teaching communication and teamwork skills with purposefully designed scaffolds. After the course, students were matched with professional, online summer internships in which they could apply the skills they learned. We detail the course structure, including the pedagogical strategies employed and the specific challenges encountered. Our experiences reveal key elements that contributed to the course's success, such as the importance of adaptive teaching methods and the need for carefully considered instructor training for teaching neurodivergent learners in software engineering. We share this report to provide guidance for educators, researchers, and advocates seeking to develop effective computing education programs that include autistic students.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
8	Delivering Round-the-Clock Help to Software Engineering Students Using Discord: An Experience Report	This experience report describes the delivery of round-the-clock helptostudentsusingDiscord(apopularmessagingandvoice/video calling platform) in a remote software engineering course. Students in the course learn full-stack web development using Ruby on Rails and PostgreSQL, and work in teams to develop web applications. Our central goal in offering round-the-clock help using Discord wasto increase the amount of help that students receive from teachers (i.e., teaching assistants and the instructor). Indeed, we found that our 24/7-Discord approach led to a considerable increase in the amount of student–teacher interaction versus the approach used previously, which emphasized in-person office hours and a question-and-answer forum in Piazza. Moreover, students from underrepresented groups in computer science interacted with teachers at a rate comparable to other students, and we received consistently positive feedback from students regarding the approach. We also made several key observations about when students tended to seek help, including that they sought help the most between 7:00 p.m. and midnight, that help seeking spiked right before deadlines, that students posted the fewest help messages on weekends, and that students posted significantly more messages during the first half of the course, which emphasized skills assignments, versus the second half, which focused on team project work.	☐	☐	☐	☐	☐	☐	☐	☐
9	https://dl-acm-org.ez137.periodicos.capes.gov.br/doi/pdf/10.1145/3510454.3517065		☑	☐	☐	☐	☐	☐	☐	☑
10	AnEmpirical Study of the Content and Quality of Sprint Retrospectives in Undergraduate Team Software Projects	The retrospective, or retro, is a fundamental component of the Agile process, widely used in both software engineering courses and industry. In a retro, teams come together at the end of a sprint to ref lect on their team's performance. We conducted an empirical study to explore three research questions concerning retros in undergraduate team projects: (1) What do students reflect on? (2) What is the quality of their reflections? (3) How do teams' retros vary in terms of content and quality? Our study analyzed a corpus of 963 statementsdocumentedintheretrosof32undergraduatesoftwareteams (n = 182 students) enrolled in four software engineering courses at two North American universities. A content analysis revealed that teams reflected most often on their work, communication, and collaboration practices. Nearly a third of teams' reflections focused on their general work practices, while nearly half focused on specific areas of the software development lifecycle—most prominently, pull requests, issues, and coding/testing/debugging. An analysis of the quality of teams' retro reflections showed that only 13% provided justification for a strategy to be stopped, continued, or started. An analysis of team-by-team results indicated significant differences in teams' retro content and quality. We compare these results to previous studies of retros in academia and industry, and consider their implications for software engineering education.	☑	☐	☐	☐	☐	☐	☐	☑
11	Decoding Student Rating Behaviors: A Comparative Study of Frequent Self and Peer Assessments in Team Projects	Team projects are commonplace in higher education, including in the domain of software engineering, as they benefit students' collaborative learning and employability skills. However, persistent challenges (e.g., free riders, dominators, and conflicts) frequently occur in teams and, as a result, undermine teamwork effectiveness. To help identify these problems, self and peer assessments have been widely adopted and practiced by instructors, and several studies have examined students' rating behavior from those assessments. However, existing research has largely been limited to the analysis of one-time—usually at the end of a project—peer and self assessments. In our work, we have been exploring a lightweight, high-frequency peer and self assessment tool that is used throughout a project. In this paper, we present our analysis of 29,844 self and peer ratings obtained from 234 students who engaged in three team projects in a single course. Based on data collected from the participants of this study, we identified several findings related to students' rating behaviors and how instructors can make effective use of students' ratings from frequent peer assessments.	☐	☐	☐	☐	☐	☐	☐	☐
12	AnExploration of How Generative AI Affects WorkflowandCollaboration in a Software Engineering Course	Howdoes Generative AI (GenAI) impact how students work and collaborate in a software engineering course? To explore this question, we conductedanexploratorystudyinaprojectbased course where students developed three versions of a systemacrossagilesprints, withunrestricted access toGenAI tools. From survey responses of 349 students, we found that the technology was used extensively with 84% of students reporting use and 90% of them finding the technology useful. Through semi-structured interviews with 24 of the students, we delved deeper, learning that students used GenAI pervasively, not only to generate code but also to validate work retrospectively, such as checking alignment with requirements and design after implementation had begun. Students often turned to GenAI as their first point of contact, even before consulting teammates, which reduced direct interpersonal collaboration. These results suggest the need for new pedagogical strategies that address not just individual tool use, but also design reasoning and collaborative practices in GenAI-augmented teams.	☐	☐	☐	☐	☐	☐	☐	☐
13	Teamwork Quality in Agile Capstone Courses: Insights from 790 Software Engineering Students	Teamworkquality (TWQ), team members'success, and team performance are critical components of effective collaboration in modern software development. As capstone courses in software engineering aim to prepare students for industry practices, understanding the factors that influence team dynamics is essential. This study investigates the relationship between team ambition levels and TWQ, team members' success, and team performance in an agile capstone course. Our study involves 133 student teams comprising 790 students across three consecutive years (2022–2024). Data collection included 1,122 responses from pre- and post-surveys. In all three years, students were grouped into teams based on ambition levels. In 2024, the course structure was changed to place less emphasis on written project reports due to the widespread adoption of AI tools like ChatGPT. The results show that ambition levels were positively correlated with TWQ, team members' success, and team performance in 2022 and 2023, with higher ambition levels generally associated with improved collaboration and outcomes. However, in 2024, there were reduced disparities between lower and higher-ambition teams, enabling more equitable success across all teams. This study highlights that while ambition levels play a role in influencing teamwork and success, structured interventions that prioritize collaboration and inclusivity have a more substantial impact. These findings suggest that moving away from ambition-based team formation and focusing on fostering high-quality teamwork processes can better prepare students for real-world agile environments.	☑	☐	☐	☐	☐	☐	☐	☑
14	WhatMakesTeam[s] Work? AStudy of Team Characteristics in Software Engineering Projects	Teaming is a core component in practically all professional software engineering careers, and as such, is a key skill taught in many undergraduate Computer Science programs. However, not all teams manage to work together effectively, and in education, this can deprive some students of successful teaming experiences. In this work, we seek to gain insights into the characteristics of successful and unsuccessful undergraduate student teams in a software engineering course. We conduct semi-structured interviews with 18 students who have recently completed a team-based software engineering course to understand how they worked together, what challenges they faced, and how they tried to overcome these challenges. Our results show that common problems include communicating, setting and holding to deadlines, and effectively identifying tasks and their relative difficulty. Additionally, we find that self-reflection on what is working and not working or external motivators such as grades help some, but not all, teams overcome these challenges. Finally, we conclude with recommendations for educators on successful behaviours to steer teams towards, and recommendations for researchers on future work to better understand challenges that teams face.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO				MOTIVO						
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
15	Which teamwork challenges do computing students face in a project-based learning course in research methods?	Courses in research methods are crucial for developing computing students' skills, such as critical thinking, analyzing practical evidence, and applying theoretical insights. However, traditional lecture-based approaches often result in low student motivation and engagement. Project-based learning (PBL) offers an alternative byinvolving students in realistic, team-based projects, yet its effects on learning processes still need to be explored. This paper critically examines the challenges of teamwork that computing students face in a PBL research methods course. Through a case study involving master's students in computer science by reviewing reflection reports and conducting semi-structured interviews, we extracted and categorized a comprehensive list of challenges using the Holz framework. Our findings indicate that introducing PBL as a teaching approach in a research methods course may result in unique challenges for computing students. Firstly, some teams may have been following the latest technologies and trends while selecting topics for their research projects, resulting in a lack of practicality. Secondly, since some teammembersmayhavebeenmoreinterested in developing software solutions instead of using a practical problem to build an empirical research project, their teams may have selected incorrect research strategies. Thirdly, some teams used an individual incremental approach instead of a collective incremental and iterative approach in their research projects, which may have negatively affected their team activities. We also proposed adding a new layer, "How to do a research project in a team?" to the existing Holz framework to address planning and collaborating challenges in the PBL-driven research methods course.	☑	☐	☐	☐	☐	☐	☑	☐
16	A Project-Based Collaboration between Software Engineering and Criminology Students: Building Applications to Understand Racial Injustice in the Criminal Justice System	https://dl-acm-org.ez137.periodicos.capes.gov.br/doi/pdf/10.1145/3510456.3514158	☐	☐	☐	☐	☐	☐	☐	☐
17	AnEmpirical Study on Usage and Perceptions of LLMs in a Software Engineering Project	Large Language Models (LLMs) represent a leap in artificial intelligence, excelling in tasks using human language(s). Although the main focus of general-purpose LLMs is not code generation, they have shown promising results in the domain. However, the usefulness of LLMs in an academic software engineering project has not been fully explored yet. In this study, we explore the usefulness of LLMs for 214 students working in teams consisting of up to six members. Notably, in the academic course through which this study is conducted, students were encouraged to integrate LLMs into their development tool-chain, in contrast to most other academic courses that explicitly prohibit the use of LLMs. In this paper, we analyze the AI-generated code, prompts used for code generation, and the human intervention levels to integrate the code into the code base. We also conduct a perception study to gain insights into the perceived usefulness, influencing factors, and future outlook of LLM from a computer science student's perspective. Our findings suggest that LLMs can play a crucial role in the early stages of software development, especially in generating foundational code structures, and helping with syntax and error debugging. These insights provide us with a framework on how to effectively utilize LLMs as a tool to enhance the productivity of software engineering students, and highlight the necessity of shifting the educational focus toward preparing students for successful human-AI collaboration.	☑	☐	☐	☐	☐	☐	☐	☑
18	Analyzing the Communication Patterns of Different Teammate Types in a Software Engineering Course Project	Effective communication is vital for the success of professional software engineering (SE) teams. As SE courses teach essential industry skills like teamwork and collaboration, ensuring effective communication becomes important in student projects. However, poor engagement from team members can lead to conflicts, uneven workloads, and diminished learning experiences. Teammate types such as Couch Potatoes, who contribute minimally, and Hitchhikers, who rely on others while taking credit, exacerbate these issues. In contrast, Lone Wolves work independently, potentially isolating themselves, while Good Teammates actively collaborate and contribute fairly, driving team success. In this study, we aimed to investigate the communication patterns of teammate types such as Couch Potato, Hitchhiker, Lone Wolf, or Good Teammate during a SE testing course. We applied Ordered Network Analysis (ONA) to the conversational data of the teams to examine the distinct communication patterns of students whose contributions were either perceived positively (e.g., Good Teammates) or negatively (e.g., Couch Potato, Hitchhiker, Lone Wolf) by their peers. The f indings reveal distinct communication behaviors across teammate types. While Good Teammates and Couch Potatoes discussed similar content, Good Teammates communicated more frequently and consistently throughout the project. Lone Wolves seldom engaged in pleasantries, reflecting a task-focused approach, whereas Hitchhikers rarely contributed substantively to technical discussions, such as pull requests, often interacted through pleasantries. These patterns emphasize the need for early interventions and communication planning to promote accountability balance and effective student collaboration during class projects.	☑	☐	☐	☐	☐	☐	☐	☑
19	External Projects and Partners: Addressing Challenges and Minimizing Risks from the Outset	Software development projects sourced from external organizations can serve as an excellent platform to help build student competencies because they often provide an environment where students can practice applying their knowledge and skills in an authentic context. However, there are many challenges and risks that can jeopardize the successful execution of such projects. In this report, we discuss some of the lessons learned about the pain points encountered by computing faculty with over a decade of experience running a software engineering studio where teams of undergraduate students work on long-term projects sourced from external partners. Our experience is based on working with a mix of project partners with a major emphasis on non-profit and community organizations and non-technical project partners. We focus on a strategy to carefully screen prospective projects to reveal possible challenges in order to avoid or minimize risks that could impact student learning outcomes.	☑	☐	☐	☐	☐	☐	☐	☑
20	"It's Not Exactly Meant to Be Realistic": Student Perspectives on the Role of Ethics In Computing Group Projects	In computing education, group projects are often seen as an opportunity for students to gain experiences similar to what they will face in the workplace. As such, it is a pressing area for ethics education: practice incorporating ethics into project-based courses might provide encouragement and scaffolding for students to consider ethics when working on "real" technologies. In this work, we provide a preliminary look at the state of ethics education in project-based computing courses via semi-structured focus groups. These focus groups revealed that while not completely ignored, ethics had a very small role in our participants' group projects, especially in software engineering courses. Furthermore, participants generally agreed that while group projects can be useful learning opportunities, they are not realistic, and this impacts their willingness to consider ethical implications while working on projects. We compiled participant feedback and ideas on how to improve ethics education in computing group projects, and present a synthesis on how this feedback can be implemented in the classroom. We also emphasize that part of simulating "real" tech work should include acknowledgment of larger structural problems and ideally training and practice in communication and argumentation.	☑	☐	☐	☐	☐	☐	☐	☑
21	Improving Software Engineering Teamwork with Structured Feedback	Teamwork is a key learning outcome for our course, Foundations of Software Engineering. However, conflicts are inevitable in teams, and if students cannot resolve conflicts, this can lead to decreased satisfaction for everyone on the team. In this experience report, we present our approach to help students deal with conflicts that can occur in team projects. Working with faculty from the School of Business, following best practices of organizational behaviour research, weinstructedstudentsonhowtoprovidehighqualitypeer feedback, and designed activities where students provided feedback to each other following these principles. After this intervention, we compared the results of our teamwork survey with the results from the previous semester. We saw a meaningful drop in teamwork problems directly after the intervention, and the effect persisted for the rest of the semester.	☐	☐	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
22	Agile Social Software Engineer: An Empirical Approach to Mitigate and Manage Social Debts in Agile Environments	Background. In agile software engineering team projects, effective collaboration and communication between members are essential for promoting agility and lean software development. Agile methodologies, which emphasize iterative development, collaboration and adaptability, are well suited to dynamic educational environments. However, agile teams in educational contexts often face non-technical challenges known as 'social debts' that can impede productivity and limit agility. These challenges, including miscommunication, cultural barriers, and team conflicts, are particularly significant in team-based learning setups where diverse teams must work together under tight deadlines. Objectives. This research aims to identify the root causes of 'community smells' that can lead to social debts within agile teaming environments and to introduce the concept of the 'Agile Social Software Engineer' role. Specifically, we define the 'Agile Social Guardian' remit as one of the main responsibilities of this role, dedicated to managing and mitigating social debts within educational and professional agile environments. T hebroader goal is to customize welfare services for large and compulsory courses to improve the student experience and enhance learning effectiveness. Study Methods. Wepresentempirical evidence on the positive impact of the Agile Social Guardian, based on a comprehensive study involving 88 agile teams and over 700 team members,aspartofanextensiveinvestigation within an educational context. Our research methodology encompasses two key phases: empirical evaluation and practical implementation of the proposed Agile Social Guardian approach within educational setup. Findings. The study introduces a novel approach to improving team collaboration and project performance in agile education, filling a gap in the existing research and offering recommendations that extend beyond classroom and training setups. We propose a taxonomy of common causes that could lead to community smells within agile teams, which may result in social debts. Furthermore, we identify the key characteristics of the Agile Social Guardian role, which are critical in both educational and professional agile environments, and provide implementation guidelines and best practices. Conclusions. This work demonstrates the significant benefits of addressing human-centric challenges, including social debts, in credit-bearing agile training. It also underscores the need for customized welfare support in educational setups, especially for large and diverse classrooms. Our findings suggest that the Agile Social Guardian has the potential to enhance both student learning experiences and team productivity in various agile contexts. Furthermore, this framework canbeadaptedtootherteam-basedlearningenvironments, offering a path towards more effective and inclusive educational practices.	☑	☐	☐	☐	☐	☐	☐	☑
23	Exploring Factors Influencing the Satisfaction of Adult Software Engineering Students with Teamwork in Distance Education	Using team-based software development assignments is a prevalent instructional strategy in software engineering (SE) education. Students utilize these development assignments as a vehicle to (co)learn SE concepts, practice problem-solving, and develop soft skills. The satisfaction of SE students with their teamwork experience in team-based assignments is an important educational and motivational factor that contributes to increased participation in future team-based projects. There is a scarcity of research on the satisfaction of SE students with teamwork in distance and online education specifically for adult learners. This study reports on a case study conducted to identify factors influencing the satisfaction of adult SE graduate students with their teamwork experiences in team-based software design and development assignments at the Open University of the Netherlands (OUNL), a distance education university for adult learners. The self-reflection reports of 29 adult SE students, aged between 25 and 35 years, documented and self-evaluated their experiences with a team-based design and development assignment within a master SE course were analyzed using an open thematic analysis approach. The analysis of the reports revealed six categories of factors that influenced the adult online SE students' satisfaction with their team-working experience, namely, the attitude of team members, communication, collaboration, team characteristics, tooling and technology, and learning. Additionally, we conducted a literature review to identify any similarities and differences between the results obtained from the literature and those derived from our study on this topic.	☑	☐	☐	☐	☐	☐	☐	☑
24	Walking the Walk of AI Ethics: Organizational Challenges and the Individualization of Risk among Ethics Entrepreneurs	Amidst decline in public trust in technology, computing ethics have taken center stage, and critics have raised questions about corporate "ethics washing." Yet few studies examine the actual implementation of AI ethics values in technology companies. Based on a qualitative analysis of technology workers tasked with integrating AI ethics into product development, we find that workers experience an environment where policies, practices, and outcomes are decoupled. We analyze AI ethics workers as ethics entrepreneurs who work to institutionalize new ethics-related practices within organizations. We show that ethics entrepreneurs face three major barriers to their work. First, they struggle to have ethics prioritized in an environment centered around software product launches. Second, ethics are difficult to quantify in a context where company goals are incentivized by metrics. Third, the frequent reorganization of teams makes it difficult to access knowledge and maintain relationships central to their work. Consequently, individuals take on great personal risk when raising ethics issues, especially when they come from marginalized backgrounds. These findings shed light on complex dynamics of institutional change at technology companies.	☑	☐	☐	☐	☐	☐	☐	☑
25	Limits and Possibilities for "Ethical AI" in Open Source: AStudy of Deepfakes	Opensource software communities are a significant site of AI development, but "Ethical AI" discourses largely focus on the problems that arise in software produced by private companies. Design, policy and tooling interventions to encourage "Ethical AI" based on studies in private companies risk being ill-suited for an open source context, which operates under radically different organizational structures, cultural norms, and incentives. In this paper, we show that significant and understudied harms and possibilities originate from differing practices of transparency and accountability in the open source community. We conducted an interview study of an AI-enabled open source Deepfake project to understand how members of that community reason about the ethics of their work. We found that notions of the "Freedom 0" to use code without any restriction, alongside beliefs about technology neutrality and technological inevitability, were central to how community members framed their responsibilities, and the actions they believed were and were not available to them. We propose a continuum between harms resulting from how a system is implemented versus how it is used, and show how commitments to radical transparency in open source allow great ethical scrutiny for harms wrought by implementation bugs, but allow harms through (mis)use to proliferate, requiring a deeper toolbox for disincentivizing harmful use. We discuss how an assumption of control over downstream uses is often implicit in discourses of "Ethical AI", but outline alternative possibilities for action in cases such as open source where this assumption may not hold.	☑	☐	☐	☐	☐	☐	☐	☑
26	Hyperion: A Generic and Distributed Mobile Offloading Framework on OpenCL	Despite the significant development of mobile device SoCs, they are still inefficient in computing computation-intensive workloads, such as high-resolution image processing and AR/VR applications. Offloading offers a promising way to leverage cloud or edge servers for acceleration, but existing offloading is limited to specific tasks or specific hardware/software platforms, resulting in significant engineering overhead. To address this problem, we focus on the underlying layer of these applications (i.e., OpenCL) and propose Hyperion, a generic and distributed mobile offloading framework built on OpenCL. To achieve high-performance distributed execution for Hyperion, we first take a deep insight into the OpenCL data structures and design regularity-aware kernel analyzer to analyze the data dependency of work-groups and identify the essential data to offload. Then, context-aware execution time predictor is proposed to estimate the computing time of a given partitioned kernel workload that is highly impacted by many runtime factors. These techniques are integrated into pipeline-enabled and network-adaptive scheduler to make scheduling decisions, which coordinates the kernel partition and workload scheduling to form pipeline processing between data transmission and distributed execution with flexible adaptability to network dynamics. Extensive experimental results demonstrate that Hyperion achieves superior performance with an average 3.80× speedup compared with the best baseline and f lexible adaptation to dynamic network conditions and available computing resources.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
27	Understanding Operational Technology Personnel's Mindsets and Their Effect on Cybersecurity Perceptions: A Qualitative Study With Operational Technology Cybersecurity Practitioners	Operational Technology (OT) is hardware and software that monitor and control industrial processes in sectors such as water and energy. OT's increased digitalisation which expanded its attack surface, heightened institutional pressures including regulation, and the evolving threat landscape have compelled companies using OTto improve their cybersecurity. Security often clashes with employees' values and hinders them from performing their primary job tasks, leading to its circumvention. OT personnel are a user group with distinctive characteristics, including the cyber-physical technology they use, and subsequently, their safety culture. Nevertheless, there's little research on how OT personnel's mindsets are formed, and in turn, how these mindsets affect their cybersecurity beliefs and behaviours. As such, we have conducted 72 interviews with OT cybersecurity practitioners across various sectors on their experiences working with OT practitioners. Our analysis demonstrates a number of factors that shape OT personnel's mindsets: namely, the prioritisation of other operational values, operational realities and challenges, and their occupational pathways. In turn, these factors lead to misperceptions around cybersecurity, specifically in two areas: technological misperceptions, and the stereotyping of occupational practices between IT and OT. Accordingly, we discuss OT cybersecurity practitioners' efforts to influence cybersecurity in these environments, and how the acknowledgment of these factors and misperceptions aids their efforts. Finally, we call for a better understanding of OT personnel's relationship with cybersecurity by proposing future research avenues.	☑	☐	☐	☐	☐	☐	☐	☑
28	SoK: Blockchain Governance	Blockchain systems come with a promise of decentralization that, more often than not, stumbles on a roadblock when key decisions about modifyingthesoftwarecodebaseneedtobemade.Inasetting where "code-is-law," modifying the code can be a controversial process, frustrating to system stakeholders, and, most crucially, highly disruptive for the underlying systems. This is attested by the fact that both of the two major cryptocurrencies, Bitcoin and Ethereum, have undergone "hard forks" that resulted in the creation of alternative systems which divided engineering teams, computational resources, and duplicated digital assets creating confusion for the wider community and opportunities for fraudulent activities. The above events, and numerous other similar ones, underscore the importance of Blockchaingovernance,namelythesetofprocessesthat blockchain platforms utilize in order to perform decision-making andconvergetoawidelyacceptteddirectionforthesystemtoevolve. While a rich topic of study in other areas, including social choice theory and electronic voting for public office elections, governance of blockchain platforms is lacking a well established set of methods and practices that are adopted industry wide. Instead, different systems adopt approaches of a variable level of sophistication and degree of integration within the platform and its functionality. This makes the topic of blockchain governance a fertile domain for a thorough systematization that we undertake in this work. Our methodology starts by distilling a comprehensive array of properties for sound governance systems drawn from academic sources as well as grey literature of election systems and blockchain white papers. These are divided into seven categories, suffrage, Pareto efficiency, confidentiality, verifiability, accountability, sustainability and liveness that capture the whole spectrum of desiderata of governance systems. We interpret these properties in the context of blockchain platforms and proceed to classify ten blockchain systems whose governance processes are sufficiently well documented in system white papers, or it can be inferred by publicly available information and software. While all the identified properties are satisfied, even partially, by at least one system, we observe that there exists no system that satisfies most properties. Our work lays out a common foundation for assessing governance processes in blockchain systems and while it highlights shortcomings and deficiencies in currently deployed systems, it can also be a catalyst for improving these processes to the highest possible standard with appropriate trade-offs, something direly needed for blockchain platforms to operate effectively in the long term.	☑	☐	☐	☐	☐	☐	☑	☐
29	Genomics Cybersecurity Concerns, Challenges, and a Modular Test Lab	Genomic data has very high value and presents its own cybersecurity challenges and concerns due to its characteristics. In order to effectively secure this valuable genomic data, it is important to understand the life cycle that creates the data, the hardware and software used, the users involved, and the challenges and concerns involved in securing genomic data. Building off of previous work done by the National Institute of Standards and Technology on NIST IR 8432, our team undertook a research review to further understand the state of cybersecurity research in the field. Then, we performed a field study by interviewing specialists and touring facilities at the HudsonAlpha Institute for Biotechnology to gain a further understanding of cybersecurity in practice for real-world genomics labs. Based on our research review and field study with HudsonAlpha, a high-level theoretical biocybersecurity test lab for genomics was designed that mimics a functional genomics lab to allow for cybersecurity testing of new tools and techniques used to protect genomic data, secure devices, and maintain the integrity of genomic data throughout its life cycle. This research can impact cybersecurity for genomic data by providing a research review, a field study, and a high-level biocybersecurity lab design in one place for future work to build off of. This work constitutes a step in taking the challenges presented by cybersecurity for genomics and creating a cybersecurity test lab that can address them.	☑	☐	☐	☐	☐	☐	☑	☐
30	Bus Factor In Practice	Bus factor is a metric that identifies how resilient is the project to the sudden engineer turnover. It states the minimal number of engineers that have to be hit by a bus for a project to be stalled. Even though the metric is often discussed in the community, few studies consider its general relevance. Moreover, the existing tools for bus factor estimation focus solely on the data from version control systems, even though there exists other channels for knowledge generation and distribution. With a survey of 269 engineers, we f ind that the bus factor is perceived as an important problem in collective development, and determine the highest impact channels of knowledge generation and distribution in software development teams. We also propose a multimodal bus factor estimation algorithm that uses data on code reviews and meetings together with the VCS data. We test the algorithm on 13 projects developed at JetBrains and compared its results to the results of the state-of-theart tool by Avelino et al. against the ground truth collected in a survey of the engineers working on these projects. Our algorithm is slightly better in terms of both predicting the bus factor as well as key developers compared to the results of Avelino et al. Finally, we use the interviews and the surveys to derive a set of best practices to address the bus factor issue and proposals for the possible bus factor assessment tool.	☑	☐	☐	☐	☐	☐	☐	☑
31	Anti-Games, Fantasy Consoles, and the Rise of Speculative Game Development on Itch.io	Amid the ongoing escalation of anti-worker and anti-player practices across the game industry, the creative practices of developers are shifting: jammers, students, hobbyists, fangamers, and other digital game "zinesters" [3] seek to reclaim game design as a personal art form. This paper examines the emergence and rise of speculative game design communities on itch.io—sites of ongoing creative resistance to corporate encroachment and exclusionary game culture. Specifically, the study documents the activities of "anti-game" jammers and fantasy console development communities on itch.io—two groups that create speculative games (that is, projects that are conceptual, arbitrary, and materially or procedurally unrealizable) to make space for personal play that is divested from mainstream engines, markets, and pretenses of game commodity: 1) Firstly, "anti-game" designers use game jams, zines, and Discord servers to develop and circulate not-games, games that refuse play, and games that cannot be played. These makers create provocative thought experiments, poems, manifestos, and other "unplayable" works that resist hyper-capitalist investment in bigger, longer, more graphically precise, live-service, metaversal, forever-games. 2) Simultaneously, developers on itch build projects for fantasy consoles—such as PICO-8 and Bitsy—and imagine hardware that is mutually-constituted, rooted in community, and decommodified. Ultimately, this paper argues that speculative design practices divest from concerns of "playability," center unencumbered participation, and allow developers to envision and participate in a post-capitalist imaginary of hardware, software, and play.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO				MOTIVO						
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
32	Cultivating a Team Mindset about Productivity with a Nudge: AField Study in Hybrid Development Teams	While there has been significant study of both individuals and teams of knowledge workers, research has focused largely on one or the other, with less focus on the interaction between the two. In this paper, we explore the tensions between the individual and their team, focusing on the choices an individual makes towards their own productivity versus their team's productivity. We developed a technology probe with a team nudge that fosters recurring reflection and prompts individuals to consider how their team helps them to be productive. We examined its impact through a longitudinal field study with 48 participants. We chose to undertake this study with software development teams as they are examples of knowledge workers who collaborate on a shared set of tasks with specific goals. Our exploration took place with hybrid development teams, which have increasingly become the norm. Our analysis of a total of 8338 hourly self-reports and 1389 daily diary entries found that the team nudge increased participants' productivity ratings and team awareness, led to participants spending more time on their own tasks, reshaped their perceptions of themselves and their team, yet, in general, did not increase team cohesion or affect well-being.	☑	☐	☐	☐	☐	☐	☐	☑
33	Design and Development of an Interactive AI Toolkit for Engaging Older Adults in AI Discussions	This paper presents the design and development of a low-cost, interactive Artificial Intelligence (AI) toolkit in the form of an animatronic 'talking' mask. The toolkit aims to democratise AI and make it more accessible, particularly for older adults, by facilitating hands-on engagement with AI technologies and engaging them in meaningful discussions about its opportunities and potential drawbacks. The toolkit is built using customisable features such as replaceable face, jaws and eyes, and modular software features like face tracking, speech recognition, speech-to-text and text-tospeech engines, and large language model (LLM). The customisable design provides a means to explore perceptions, raise awareness and encourage healthy discussions about different facets of AI, especially amongst technologically-challenged populations. It promotes co-design opportunities by leveraging its customisability to allow for adaption of its features to suit the technical abilities and preferences of the audience. Designed with an open-source approach, the project also encourages collaboration among makers and developers, fostering community-driven enhancements and adaptations. This accessibility empowers others to build upon the design, contributing to its evolution as a versatile tool for education and outreach. Evaluation of the platform with 3 distinct user groups highlights its potential to demystify AI technologies and engage diverse audiences in a conversation about the challenges and opportunities they present. This paper concludes with the identification of future opportunities for this research, including the use of co-creation methodologies with end users to extend the work to suit their needs and provide a public engagement framework for understanding emerging technologies.	☑	☐	☐	☐	☐	☐	☑	☐
34	Beyond "Awareness": If We Teach Inclusive Design, Will Students Act OnIt?	Motivation: Many university CS programs have begun teaching various types of CS-related societal issues using approaches such as ethics, Responsible CS, inclusive design, and more. However, some recent research suggests that, although these programs have been able to teach awareness, students often fail to act upon this awareness. To address this problem, University X's CS program tried an unusual approach—integrating hands-on inclusive design skills in small ways across all four years of the CS major. But did it work? That is, did the students who experienced this change across the major actually build more inclusive technology than the students who did not experience it? Objectives: This paper aims to answer this through addressing two research questions: (RQ1): Did students who learned inclusive design across the curriculum act to create more inclusive software? (RQ2): How did inclusivity (or lack thereof) manifest in students' projects? Method: To investigate these RQs, we conducted a case study of 22 term-long CS projects built by 22 teams consisting of a total of 92 3rd- and 4th-year CS students. Half of the student teams had experienced courses that had integrated inclusive design and the other half had not. The inclusive design elements University X taught were those of the GenderMag inclusive design method, so evaluating the students' term-long projects was done by GenderMag experts—industry-experienced UX and Software professionals with real-world GenderMag experience. Results: The inclusiveness of students' projects was higher Post-GenderMag, with fewer reports of inclusivity bugs and higher inclusivity ratings. Experts' evaluations also revealed the ways in which bias (e.g. bias against risk-averse users) and inclusion appeared in students' projects. Implications: We believe this to be the first published evidence that compares student-built technology's inclusiveness before vs. after they have been taught inclusive design. These positive results suggest that teaching inclusive design across the curriculum can impact students beyond simply heightening awareness—moving them to act upon this new understanding by building technology that more inclusively serves a wider spectrum of society	☑	☐	☐	☐	☐	☐	☐	☑
35	https://dl-acm-org.ez137.periodicos.capes.gov.br/doi/pdf/10.1145/3524487.3527356		☑	☐	☐	☐	☐	☐	☑	☐
36	https://dl-acm-org.ez137.periodicos.capes.gov.br/doi/pdf/10.1145/3528231.3528358		☐	☐	☐	☐	☐	☐	☐	☐
37	Exploring LLM-Based Agents for Root Cause Analysis	Root cause analysis (RCA), a critical part of the incident management process, is a demanding task for on-call engineers, requiring deep domain knowledge and extensive experience with a team's speci c services. Automation of RCA can result in signi cant savings of time, and easetheburdenofincident managementonon-call engineers. Recently, researchers have utilized Large Language Models (LLMs) to perform RCA, and have demonstrated promising results. However, these approaches are not able to dynamically collect additional diagnostic information such as incident related logs, metrics or databases, severely restricting their ability to diagnose root causes. In this work, we explore the use of LLM based agents for RCA to address this limitation. We present a thorough empirical evaluation of a ReAct agent equipped with retrieval tools, on an out-of-distribution dataset of production incidents collected at Microsoft. Results show that ReAct performs competitively with strong retrieval and reasoning baselines, but with highly increased factual accuracy. We also conduct a case study with a team at Microsoft to allow the ReAct agent to interface with diagnostic services that are used by the team for manual RCA. Our results show how agents can overcome the limitations of prior work, and considerations for implementing such a system in practice.	☑	☐	☐	☐	☐	☐	☑	☐
38	More Sustainable than Sustainable	Prominent software executives have vocally proclaimed that tech such as AI and space exploration will resolve earthlings' sustainability crisis. Common to these proposals is a paradigm of everincreasing human demands met by ever-increasing technological scope, capacity and surveillant efficiency. Advocates (Diamantis 2016, Andreessen 2023) label this paradigm abundance. Software practitioners, who collaborate to create the functionality on which these advocates' platforms are built, also employ a notion of abundance — often in their public commentary, and especially consistently in their conventional practices — but of a very different sort. T his paper uses content analysis of more than two hundred public statements (microblogs, blog posts and publications) by software practitioners and executives, plus 17 published assessments of AI assistant software's impact on development work, alongside praxiographic analysis (Mol 2002) of conventional cross-disciplinary software team practices (such as debugging, user-centered design and sprint retrospectives), to discern an ecosystemic understanding of abundance enacted in software team practice. It then contrasts the two versions of abundance and suggests the potential benefit of the one hidden at the heart of collaborative software practice to the project of civil and environmental sustainability on earth.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
39	Investigating Reflection in Undergraduate Software Development Teams: An Analysis of Online Chat Transcripts	Metacognition is widely acknowledged as a key soft skill in collaborative software development. The ability to plan, monitor, and reflect on cognitive and team processes is crucial to the efficient and effective functioning of a software team. To explore students' use of reflection—one aspect of metacognition—in undergraduate team software projects, we analyzed the online chat channels of teams participating in agile software development projects in two undergraduate courses that took place exclusively online (n = 23 teams, 117 students, and 4,915 chat messages). Teams' online chats were dominated by discussions of work completed and to be done; just two percent of all chat messages showed evidence of reflection. Afollow-up analysis of chat vignettes centered around reflection messages (n = 63) indicates that three-fourths of the those messages were prompted by a course requirement; just 14% arose organically within the context of teams' ongoing project work. Based on our f indings, we identify opportunities for computing educators to increase, through pedagogical and technological interventions, teams' use of reflection in team software projects.	☑	☐	☐	☐	☐	☐	☐	☑
40	team software projects, software engineering, software engineering education, metacognition, reflection, content analysis, agile	Microtask programming enables software engineers such as freelancers and part-time employees to contribute to software projects even when they can not spend much time on them. It decomposes software design into small, self-contained specifications. The decomposed specifications enable them to complete implementation and review task in a short time. In this paper, we empirically investigate the time required for software engineers to complete microtasks in an industrial setting and explore their perceptions of microtask programming by investigating two industrial projects using it. The projects were carried out in different companies and differed in the employment of the engineers. One contracted 9 freelancers, and the other asked for 8 part-time contributions from employees at work on other projects. We conducted a survey and a focus group with the engineers. Based on the development data of the case studies, we found that almost all microtasks were completed in less than four pomodoro repetitions, namely about two hours in the pomodoro technique. These data shows that engineers who cannot work full-time on a project can undertake microtasks if they can spare one-third of their work day. We also examine how engineers who are employees experience microtask programming similarly and differently from freelancers.	☑	☐	☐	☐	☐	☐	☑	☐
41	Integrating Small-scale Autonomous Vehicles in CS Education: An Experience Report	Teaching software systems engineering is neither effective, nor inspiring if students cannot practice the conveyed theory. We report onahands-onapproachthatclosesthegapbymeansofoff-the-shelf and in-house realistic miniature models of autonomous vehicles. It is innovative that the infrastructure extends beyond the widelyused solutions, to digitally controlled cars and trains, with open hardware and extendable sensory/actuating capabilities (cameras, opto-sensors, proximity sensors, LEDs, displays) operating in realistic mini-environments, including line-marked roads, railways, traffic-lights and-signs, tunnels and railroad switches. For many years already, we have been using these vehicles to structurally teach physical computing (300 undergraduates yearly) and systems testing (12 graduates yearly), and for incidental individual research projects. Recently, we started two initiatives that stimulate a more dynamic know-how transfer across generations. The first is a better scalable undergraduate capstone project, where groups of students work on a 'hot' research topic in autonomous driving. The second motivates student teams to go the extra mile in an international intelligent car race competition. Evaluations showed that although unusual for a non-engineering curriculum, 'playing' with autonomous vehicles is an excellent strategy to discover the interaction of software with hardware and environment, to better consolidate existing knowledge on programming and testing, and to explore new fields, such as AI-based computer vision, navigation and safety, adding in all cases more fun and motivation. We share the design of the scaffolding and teaching initiatives, together with lessons learned which will hopefully inspire other educators in shaping engaging and future-proof CS curricula.	☑	☐	☐	☐	☐	☐	☐	☑
42	GoodFences Make GoodNeighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells	Software development is de facto a social activity that often involves people from all places to join forces globally. In such common instances, project managers must face social challenges, e.g., personality conflicts and language barriers, which often amount literally to "culture shock". In this paper, we seek to analyze and illustrate how cultural and geographical dispersion—that is, how much a community is diverse in terms of its members' cultural attitudes and geographical collocation—influence the emergence of collaboration and communication problems in open-source communities, a.k.a. community smells, the socio-technical precursors of unforeseen, often nasty organizational conditions amounting collectively to the phenomenon called social debt. We perform an extensive empirical study on cultural characteristics of GitHub developers, and build a regression model relating the two types of dispersion—cultural and geographical—with the emergence of four types of community smells, i.e., Organizational Silo, Lone Wolf, Radio Silence, and Black Cloud. Results indicate that cultural and geographical factors influence collaboration and communication within open-source communities, to an extent which incites—or even more interestingly mitigates, in some cases—community smells, e.g., Lone Wolf, in development teams. Managers can use these findings to address their own organizational structure and tentatively diagnose any nasty phenomena related to the conditions under study.	☑	☐	☐	☐	☐	☐	☑	☐
43	Industry Experiences with Large-Scale Refactoring	Software refactoring plays an important role in software engineering. Developers often turn to refactoring when they want to restructure software to improve its quality without changing its external behavior. Small-scale (floss) refactoring is common in industry and is often performed by a single developer in short sessions, even though developers do much of this work manually instead of using refactoring tools. However, some refactoring efforts are much larger in scale, requiring entire teams and months or years of effort, and the role of tools in these efforts is not as well studied. In this paper, we report on a survey we conducted with developers to understand large-scale refactoring and its tool support needs. Our results from 107 industry developers demonstrate that projects commonly go through multiple large-scale refactorings, each of which requires considerable effort. Our study finds that developers use several categories of tools to support large-scale refactoring and rely more heavily on general-purpose tools like IDEs than on tools designed specifically to support refactoring. Tool support varies across the different activities, with some particularly challenging activities seeing little use of tools in practice. Furthermore, our analysis suggests significant impact is possible through advances in tool support for comprehension and testing, as well as through support for the needs of business stakeholders.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO				MOTIVO					
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
44	The Effect (or Lack Thereof) of Spatial Skills Training in a Mid-Major Computing Course	Background and Context. Spatial skills have been shown to correlate with success in introductory computing courses. Fortunately, spatial skills can be trained to improve outcomes for all students, both in terms of greater spatial skills but also in terms of achievement in the computing classroom. However, less is known about courses beyond the introductory computing sequence. Objectives. Our work sought to understand the impact of spatial skills training on both spatial ability and computer science performance in a mid-major computer science course. We further exploredwhatfactorsinfluencetheoutcomesattheendofatraining intervention, including the effort students applied to the training. Methods. This paper describes a quasi-experimental study of spatial skills training in a required software engineering course in the middle of an undergraduate computing degree program. One section of the course (n=31) received spatial skills training while another section of the course (n=46) received more courserelated activities to control for time-on-task. Each section took pre- and post-tests in spatial skills and computing knowledge and additional demographic information wasgathered.Thespatialskills training artifacts were further graded and analyzed for effort and correctness. Findings. Our statistical analyses show no significant differences in the pre- and post-scores for the treatment group. The primary predictor of post-test spatial skills score was pre-test spatial skills score, regardless of the condition. Our analyses do show that the more effort students put into the spatial skills training, the more likely they are to improve their spatial skills, though not necessarily for low spatial skills students. We also find evidence of spatial skills training mitigating gaps in spatial abilities between men and women students. Implications. Our work indicates the importance of timing and motivation when implementing spatial skills training. These training activities will not be as effective at all points in a computing major, nor is it significantly beneficial to apply the training to all students in a course. Our findings highlight the need for further research on where and when spatial skills training can be the most effective for computer science undergraduate students and how it can be feasibly implemented within the constraints posed by a program of study.	☑	☐	☐	☐	☐	☐	☐	☑
45	WorkPackets: A New Abstraction for GC Software Engineering, Optimization, and Innovation	Garbage collection (GC) implementations must meet efficiency and maintainability requirements, which are often perceived to be at odds. Moreover, the desire for efficiency typically sacrifices agility, undermining rapid development and innovation, with unintended consequences on longer-term performance aspirations. Prior GC implementations struggle to: i) maximize efficiency, parallelism, and hardware utilization, while ii) correctly and elegantly implementing optimizations and scheduling constraints. This struggle is reflected in today's implementations, which tend to be monolithic and depend on coarse phase-based synchronization. This paper presents a new design for GC implementations that emphasizes both agility and efficiency. The design simplifies and unifies all GC tasks into work packets which define: i) work items, ii) kernels that process them, and iii) scheduling constraints. Our simple insights are that execution is dominated by a few very small, heavily executed kernels, and that GC implementations are high-level algorithms that orchestrate vast numbers of performance-critical work items. Work packets comprise groups of like work items, such as the scanning of a thread's stack or the tracing of a single object in a multi-million object heap. The kernel attached to a packet specifies how to process items within the packet, such as how to scan a stack, or how to trace an object. The scheduling constraints express dependencies, e.g. all mutators must stop before copying any objects. Fully parallel activities, such as scanning roots and performing a transitive closure, proceed with little synchronization. The implementation of a GC algorithm reduces to declaring required work packets, their kernels, and dependencies. The execution model operates transparently of GC algorithms and work packet type. We broaden the scope of work-stealing, applying it to any type of GC work and introduce a novel two-tier work-stealing algorithm to further optimize parallelism at fine granularity. Weshowthe software engineering benefits of this design via eight collectors that use 23 common work packet types in the MMTk GC framework. We use the LXR collector to show that the work packet abstraction supports innovation and high performance: i) comparing versions of LXR, work packets deliver performance benefits over a phase-based approach, and ii) LXR with work packets outperforms the highly-tuned latest (OpenJDK 24), state-of-the-art G1 garbage collector. We thus demonstrate that work packets achieve high performance, while simplifying GC implementation, making them inherently easier to optimize and verify.	☑	☐	☐	☐	☐	☐	☑	☐
46	Effectiveness of Video-based Training for Face-to-face Communication Skills of Software Engineers: Evidence from a Three-year Study	Objectives. Communication skills are crucial for effective software development teams, but those skills are difficult to teach. The goal of our project is to evaluate the effectiveness of teaching face-to-face commu- nication skills using AVW-Space, a platform for video-based learning that provides personalized nudges to support student's engagement during video watching. Participants. The participants in our study are second-year software engineering students. The study was conducted over three years, with students enrolled in a semester-long project course. Study Method. We performed a quasi-experimental study over three years to teach face-to-face com- munication using AVW-Space, a video-based learning platform. We present the instance of AVW-Space we developed to teach face-to-face communication. Participants watched and commented on 10 videos and later commented on the recording of their own team meeting. In 2020, the participants (n = 50) did not receive nudges, and we use the data collected that year as control. In 2021 (n = 49) and 2022 (n = 48), nudges were provided adaptively to encourage students to write more and higher-quality comments. Findings. The findings from the study show the effectiveness of nudges. We found significant differences in engagement when nudges were provided. Furthermore, there is a causal effect of nudges on the interaction time, the total number of comments written, and the number of high-quality comments, as well as on learning. Finally, participants exposed to nudges reported higher perceived learning. Conclusions. Our research shows the effect of nudges on student engagement and learning while using the instance of AVW-Space for teaching face-to-face communication skills. Future work will explore other soft skills, as well as providing explanations for the decisions made by AVW-Space.	☑	☐	☐	☐	☐	☐	☐	☑
47	Early Career Software DevelopersAre You Sinking or Swimming?	Background: Newbies in their early stage of software engineering careers suffer from unfitting task assignments, unclear job expectations, and insufficient communication with managers frequently, which leads to personal frustration, unsatisfactory team performance, and low employee retention. Goals: The goal of this research is to investigate new software developers' "sink or swim" early career experience from the following four dimensions: job assignment, newbie-manager pairing, job satisfaction, and thoughts and suggestions. Methodology: To achieve our research goal, we conducted an empirical study by distributing an online questionnaire that includes both qualitative and quantitative questions. Results: There are several factors contributing to a "sink or swim" early career experience, such as unclear about what to do, who to report to, lack of communication, and vague expectations, posing negative impacts on both individuals and the organization. In addition, we also propose a new communitysmellinourpaper-Newbie Sink or Swim, based on our investigation. Conclusions: The early stage is critical to software developers' careers. A failing start phase has detrimental effects on software developers and development teams. Our study empirically examines software developers' early careers from various aspects, providing deeper insights into how to build a more supportive and productive working environment for entry-level developers in the software community.	☑	☐	☐	☐	☐	☐	☐	☑
48	Project-Sized Scaffolding for Software Engineering Courses	Students can often graduate with a degree in computer science without working with legacy code bases, yet when they join the workforce they will almost certainly work on an existing project with thousands, if not millions of lines of existing code. In order to give students a realistic experience without overwhelming them, we added scaffolding to an existing open source project and used it in our third year software engineering course. We asked students to complete aseries of 5tasks, from bugfixingtofeatureaddition, with this scaffolded project. Our scaffolding consisted of enhanced documentation, demonstration videos, compilation videos, enhanced task descriptions, and hints for task completion. After running this course project we collected feedback via a survey (n=87) and a small focus group (n=7). We found that students appreciated the realistic experience, but that they recommend further scaffolding, especially within source code, to better balance between difficulty and learning.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
SCOPUS										
49	On the Impact of Grading on Teamwork Quality in a Software Engineering Capstone Course	Every semester, we deliver a capstone course on software engineering where students undertake a real-world project in three iterations. By the end of each iteration, students are graded in several dimensions: software quality, project management, and peer assessment. The latter is the only grade assigned individually; therefore, students who are penalized by their teams (e.g., for being perceived as low contributors to the team effort) are not severely affected. This results in little incentive for improvement, which potentially jeopardizes the overall quality of the project outcome. Envisaging to promote team cohesion, we devised a new grading rule: if the peer assessment of a student is lower than a threshold, that would be their final grade in the iteration. This paper reports the results of studying the effectiveness of the proposed rule. We recorded peer assessments over six consecutive semesters: (1) the first three as the baseline measure; (2) the semester where we introduced the new grading rule; and (3) the following two semesters, as a contrast. When the rule was first introduced, peer assessments resulted low and heavily spread at the beginning, but they consistently improved toward the end of the semester. When the instructional team already trusted the rule and explicitly emphasized its application, peer assessments consistently grew along the semester but resulted in fewer outliers. The study results show that exposing peer assessments earlier on helps promote team reflection. They also made evident the positive impact of teamwork for producing quality products in a software engineering capstone course. © 2023 Elsevier B.V., All rights reserved.	☐	☐	☐	☐	☐	☐	☐	☐
50	The Influence of Software Engineering Education on Developing Teamwork Skills in Both Genders	Teamwork is recognized as a highly valuable skill for any software engineer. In computer science programs it is typically developed through project-based courses, though these courses often do not adopt a gender-oriented approach. Women are underrepresented in the software industry, and there is limited knowledge about their teamwork competencies. While some studies compare teamwork skills between men and women, few have explored the impact of education on the development of this skill, particularly in relation to different aspects of teamwork. By comparing teamwork quality between genders at different program levels, we intend to gain valuable insights about students' initial teamwork skills and the effect of software engineering education over time. To do this, we have carried out an empirical study of two project-based software engineering courses in our Computer Science and Engineering program: an introductory software engineering course and a capstone course. In each case we evaluate different aspects of teamwork by applying the same peer assessment instrument along 13 semesters. Our study revealed no significant gender-based disparities within the courses under study, indicating that such differences are absent both in students' initial abilities and final competencies. What we did observe, however, is that women's teamwork assessments are more homogeneous over time than the men's, where we see significant improvements in Attitude, Motivation, Coordination and Communication between courses. Our results lead us to believe that, it takes men longer to reach the same capacity for working in teams than women, but in the end, students of both genders reach similar teamwork capabilities in all teamwork quality dimensions. Therefore, women have just as much potential to excel as software engineers as men, especially when it comes to teamwork. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
51	SIMULATING AN ENGINEERING WORKPLACE: A NEW APPROACH TO PROTOTYPE-BASED TEAM PROJECT	This paper documents the remote management of a first-year foundations of engineering course with special focus on students' learning by completing a prototype-based project in an online course. The COVID-19 pandemic brought on unprecedented challenges to the teaching and learning communities around the world. Educators made purposeful changes in their teaching approaches, shifting rapidly from in-person to online mode of instruction. This study documents a project-based course that adopted an asynchronous mode of instruction as a part of the general engineering curriculum at a large Southeast university in the United States during the pandemic. This asynchronous course - through implementing necessary changes and adaptations - simulated the experience of a cross-border engineering workplace. The course content focuses on engineering design and problem-solving, physical prototyping, simulated data collection and analysis, contemporary software tools, and professional practices and expectations (e.g., communication, teamwork, and ethics). Learning activities are designed to introduce students to the types of work that engineers do daily and to challenge students' knowledge and abilities as they explore the different elements of engineering by completing an aesthetic wind turbine project. Our paper reports on the development of the course site as informed by recent national developments in scholarship and practice for online teaching and learning. The principles of course design alignment as well as instructor presence and learner interaction as suggested by these national standards are discussed. Further, the study records strategies adapted to enable students to complete a successful prototype-based project working in geographically distributed and virtual, international teams. © 2023 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☑	☐
52	Students' Perceptions of Problem-Based Learning in Multidisciplinary Groups When Seeking to Solve an Engineering Grand Challenge	This paper presents findings from a small-scale research study eliciting students' perceptions of benefits and challenges of working in interdisciplinary groups to solve an engineering challenge using problem-based learning. Penultimate and final year undergraduates and postgraduate MSc students in the School of Engineering and Physical Sciences at a Scottish university, studying Robotics, Mechanical, Chemical, Electrical and Software Engineering worked in interdisciplinary groups of five on a project to provide solutions to the United States National Academy of Engineering Grand Challenges (NAEGC). Students were surveyed twice, using closed and open questions before and towards the end of the project. Data were analysed using a thematic approach. Findings showed that most students saw benefits to problem-based working with students from other disciplines, citing increased awareness of approaches, future 'real world' professional preparation and efficiency in problem solving. However, challenges around scheduling meetings and concerns around cross-discipline collaboration indicate that universities should provide training for students before undertaking such problem-based projects, to ensure maximum educational benefits. In addition, greater emphasis needs to be put on students' awareness of the added benefits of development of the 'soft skills' needed for future professional practice. © 2022 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
53	Improving Software Engineering Teamwork with Structured Feedback	Teamwork is a key learning outcome for our course, Foundations of Software Engineering. However, conflicts are inevitable in teams, and if students cannot resolve conflicts, this can lead to decreased satisfaction for everyone on the team. In this experience report, we present our approach to help students deal with conflicts that can occur in team projects. Working with faculty from the School of Business, following best practices of organizational behaviour research, we instructed students on how to provide high quality peer feedback, and designed activities where students provided feedback to each other following these principles. After this intervention, we compared the results of our teamwork survey with the results from the previous semester. We saw a meaningful drop in teamwork problems directly after the intervention, and the effect persisted for the rest of the semester. © 2024 Elsevier B.V., All rights reserved.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
54	Status indicators in software engineering group projects	A segment of studies on group structure and performance in software engineering (SE) project-based learning (PjBL) have focused on roles, including studies that use Belbin team roles and studies that address problematic roles such as social loafing. The present study focuses on the status, which is basically missing in SE PjBL studies, although relating to roles. The study investigates the aspects that students identified as indicators of rising or declining status in their project groups. The status theory was utilized as the framework that motivated the research and on which the results were reflected. An inductive qualitative content analysis was applied to learning reports in which students reflected on their statuses. The indicators of rising status included technical know-how, commitment, management responsibility, and idea ownership, while also group-level attributes such as a caring atmosphere and joint responsibility. The indicators of a declining status included aspects that appear as counterparts of rising status indicators, while also more refined aspects such as no one willing to be a leader or study background. The results are concluded to provide material for educating students about intra-group relations and promoting self-regulation for fruitful collaboration in groups. The authors believe that the results also initiate further PjBL research in which status theory can be utilized. © 2023 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
55	A systematic mapping study on group work research in computing education projects	Context: For developing students' group- and teamwork skills needed in the team-oriented work environments of the software industry, the role of project-based learning is considered central. Yet there does not appear to be a proper mapping of the current group work research in the computer science project education literature. Thus, the current state of group work research in the research area is somewhat unknown. Objective: This study aims to form an overview of how research has addressed students' group work in the field of computer science (CS) and software engineering (SE) to identify research gaps as well as suitable topics for more detailed literature reviews. Methods: A systematic mapping study was used to investigate how group work in tertiary education has been undertaken in the literature during the past decade. Results: Based on the selected papers, the most investigated group work areas were related to the assessment of groups, group formation, communication, and cooperation. The research appeared to be quite narrowly focused on a few areas. Most of the papers were experience or evaluation research. A case study using interviews or questionnaires to gather data from a single course was the most representative type of study. The papers were mainly published in scientific conferences. The use of theoretical frameworks was limited, with a focus on a few established frameworks. Tuckman's group development theory was the predominant framework, while other commonly used concepts and theories include social loafing, Kolb's learning style theory, and the Big Five personality traits model. Conclusion: Out of 7515 papers screened, 225 were deemed eligible and analyzed. We conclude a need for more focused group work research in CS/SE student projects, in which education is inspected from particular perspectives. This would create identifiable lines of research and structure the research area. Relatedly, we suggest that the underused theoretical frameworks can inspire important research: group interventions would benefit from a socially shared regulated learning perspective, explicit use of justice theories would improve theoretical understandings of group behavior, and transactional distance would help analyze how students adopt a software process. Moreover, the research area could be precipitated by novel theoretical perspectives. For practitioners, those implementing a group project course can benefit from a large amount of literature on assessment and group formation, which are issues on which the teacher must take a position. We also include a lessons-learned summary for teachers. Generally, the present results outlining the field in a structured way can facilitate research-based teaching. © 2023 Elsevier B.V., All rights reserved.	☑	☐	☐	☑	☐	☐	☐	☐
56	The impact of unequal contributions in student software engineering team projects	Unequal distribution of work is a common problem in student team projects, undermining learning objectives and reducing student satisfaction with teamwork. More is needed to be known about the impact of unequal distribution of work on the performance of student software engineering teams and their members and the relationship between objective measures of unequal contribution and team-perceived unequal contribution. A greater understanding of these issues allows for targeted and personalised responses to these behaviours. We investigated several aspects of unequal contribution in student software engineering teams. We measured inequality of contribution using Git data from student software engineering teams, source code quality using code analyser SonarQube, and the team performance using grades. According to our results, most students under-contributed to their teams, and at least half the teams in each assignment had low equality of contribution or extreme inequality of contribution. Individual contribution styles did not strongly persist between modules (one-semester-long software engineering courses). There were no consistent associations between inequality of contribution in the student teams and performance or code quality. When the contribution of the least active team member was less than 14% of their fair share, teams were likely to perceive an unequal distribution of contributions. © 2023 Elsevier B.V., All rights reserved.	☐	☐	☐	☐	☐	☐	☐	☐
57	Relating team atmosphere and group dynamics to student software development teams' performance	Context: While the software engineering community (i.e., those involved with engineering software) is constantly in search of insights into team atmosphere and group dynamics and the way these issues impact team performance, little opportunities typically exist to explore this issue. Student projects offer an opportunity for us to understand these issues, and particularly if these students are on the verge of leaving university for post-study work and using similar practices to those used in industry. Objective: We explore a range of student software development projects' data and students' open-ended responses to five group dynamics categories: communication, time management, commitment, problem analysis and solving, and initiative and involvement. Method: We analyse both quantitative and qualitative data to study the variation in group dynamics across teams developing different software and how these variations correlated with team satisfaction. We also explore the group dynamics themes that evolve from students' open responses in relation to the five categories. Furthermore, we relate the prevalence of the themes to various software development performance metrics, before exploring the opportunity of predicting an optimum team dynamics. Results: We observe variations in the way different teams work, but higher performing teams also committed more to their projects. Various group dynamics themes were evident among functional teams, and specific patterns were more pronounced when teams were productive. Further, while there is no specific group dynamics pattern that predicts project success, successful teams were most organised and reflective. Conclusion: Competence may set the tone for positive group dynamics and team performance. Also, an achievement-driven orientation is as important as the soft skills and interpersonal aspects. © 2023 Elsevier B.V., All rights reserved.	☐	☐	☐	☐	☐	☐	☐	☐
58	Examining Teamwork: Evaluating Individual Contributions in Collaborative Software Engineering Projects	Collaborative learning and group projects are integral to Software Engineering (SE) education as they help prepare students for professional environments where teamwork and collective problem-solving are crucial. Accurately assessing individual contributions within the group project setting remains a significant challenge. Traditional assessment methods often fail to distinguish between individual efforts and collective achievements, particularly in team-based assignments prevalent in SE courses. This paper explores how several factors potentially affect individual participation in group programming projects. We compare metrics for evaluating individual contributions and examine the effect of group formation strategies on collaborative practices. Our data consist of the collaborative artifacts of 77 teams and peer evaluation responses from 154 students across two semesters of a graduate-level software engineering course, providing insights into several research questions, including the effectiveness of different metrics for estimating individual contributions, the distribution of programming tasks among students, and the impact of group composition on task distribution and peer evaluations. © 2025 Elsevier B.V., All rights reserved.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
59	Leadership Teaching in Agile Software Engineering: A Systematic Mapping	The software industry is characterized by an environment of uncertainty, high volatility, and constant change. This context has shaped the industry, its components, and actors, generating methodologies capable of meeting both market expectations and software development requirements. Among these methodologies, agile has been the most widely adopted - it provides teams with an overarching set of practices to manage software development project requirements while maintaining flexibility to incorporate changes that the business environment presents. However, professionals equipped with the necessary skills to work in these Agile teams, often referred to as "soft skills", pose a challenge for universities to teach. From this set of skills, leadership, in particular, has recently garnered attention from the software engineering education community but remains an open research opportunity. In this context, this work aims at creating a body of knowledge regarding leadership teaching in agile software engineering. To achieve this goal, we conducted a systematic mapping. From a selection of 27 studies, our results provide indications that: (i) leadership in software engineering education is typically defined as part of teamwork, shared among team members, associated with Scrum, and applied to provide students with experience in group representation; and (ii) it is usually taught through Scrum, involving active learning methodologies (such as problem-based learning) and employing real projects either sourced from external partners or designed to solve real-world problems. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☑	☐	☐	☐	☐
60	Software engineering team project courses with industrial customers: Students' insights on challenges and lessons learned	Team project courses in software engineering allow students to apply their acquired disciplinary knowledge while developing essential skills needed to work in the software industry. This paper examines the challenges and lessons learned by students in two team project courses involving industrial customers. The first course involves small teams and less complex project, whereas the second course, has larger teams and more complex projects. Using thematic analysis, we analyzed 158 reports submitted by two cohorts of students across two successive team project courses. As per our findings most challenges and lessons learned pertain to soft skills, such as teamwork, working in remote and hybrid setting, and collaboration with industrial customers. The results show that challenges and lessons learned evolve as students progress to the second team project course, for example, managing changes and addressing individual skill gaps were more pronounced in the first project course, while students reported greater coordination, communication, and contribution issues in the second team project course. The alignment between the challenges faced and the lessons learned suggests that addressing challenges in teamwork, collaborating with industrial customers, and working in hybrid or remote settings helped students develop effective strategies to mitigate these challenges. This process offers a valuable learning experience for the students, enriching their professional growth. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
61	Improving Teamwork in Software Engineering Projects in Higher Education	In computing science (CS) education, software programming and software engineering subjects are parts of core CS subjects where students learn various programming languages and software development life cycle (SDLC). With the aim of equipping students with the necessary project skills and experiences essential for workplaces, Institutes of Higher Learning (IHL) have increasingly adopted the educational practice of bringing real-world team-based projects in the curriculum. Such software projects commonly support the use of Agile process practices, which place more emphasis on communications and teamwork among students in a team, stakeholders and industry customers. This could be manageable for small scale projects where student teams are in their learning journey with active and continuous communications with customers, while developing the necessary cohesion within the team to facilitate incremental deliverables of their software products. However, teamwork remains a bottleneck that is difficult to be consistently realized among members in group projects. It is not easy to motivate and measure all team members working towards the common goals or complete project tasks on time. As such, this paper proposes the Weekly Team Assessment (WTA) framework to continuously encourage teamwork throughout the software project development duration. We also propose the settings of dedicated leadership and constructive feedback iteratively in the WTA framework. The proposed methods have been applied and evaluated through the experiments in two software engineering modules, Professional Software Development and Team Project, involving 50 sophomore CS students who are divided into control group and experimental group equally. Experiment results are analyzed and discussed in this paper. Positive outcomes are observed from the experiment where about 72% of participants appreciate the proposed methods in improving their teamwork in group projects. The degree of effectiveness of the proposed methods is further evaluated from three aspects in the experiment. © 2024 Elsevier B.V., All rights reserved.	☐	☐	☐	☐	☐	☐	☐	☐
62	Agile Software Engineering Capstone Courses: Exploring the Impact of Gender	The wide adoption of agile methodologies in software development necessitates an educational approach that prepares students for industry practices. This study aims to explore an agile capstone course, focusing on student learning experiences in teamwork while examining the gender dynamics affecting team roles and engagement. We conducted a comprehensive study of the capstone course "Software Engineering with Project Work" at the University of Oslo, analyzing 27 presentations and interviewing 10 students and two teaching assistants. The course is designed to immerse informatics students of different specializations in agile methodologies and collaborative project work. Our findings suggest that the students learn essential collaboration and technical skills. However, during their final presentations, significant gender differences emerged in the topics discussed: women predominantly spoke about design and process, whereas men focused more on technical aspects such as architecture, technical solutions, and algorithms. Our results underscore the need for educational strategies that promote inclusivity and equal participation in agile software development courses. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
63	The Role of Team Composition in Agile Software Development Education: A Gendered Perspective	Team composition is a critical factor influencing collaboration within agile software development. This study investigates the impact of gender distribution on teamwork quality in software engineering capstone courses. We examined the experiences of 240 students organized into 40 teams during an agile project course. We analyzed two surveys, one conducted before team composition and one at the end of the project work. As much as 91% of the students chose to use the practice of conducting stand-up meetings in their project work and the majority were satisfied with the practice. Further, our analysis reveals that while women tend to engage more in design and men in programming, an increase in the proportion of women within a team correlates with a higher involvement of women in programming tasks. Our findings highlight gender differences in perceptions and experiences related to project involvement in agile software engineering courses. © 2024 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
64	Managing Group Projects in Undergraduate Computing	This panel convenes four educators, each from different institutions and each with experience managing group projects. Their expertise spans topics including: peer assessment and peer evaluation; entrepreneurship; transdisciplinarity; internationalisation; inclusivity; social values; educational technology and tools; feedback and feed-forward; peer rating; free-riders; as well as blended learning; and post-pandemic online discourse. They reflect on the delight of seeing students collaborate to deliver meaningful projects as well as the challenges posed by disengaged students. They also explore a common theme of discordance inherent to teamwork and systems to support student communities. © 2023 Elsevier B.V., All rights reserved.	☑	☐	☐	☑	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
65	Teamwork in agile software development: A mixed-method study of gender diversity and collaboration dynamics	Context: Teamwork is an important aspect of agile software development, and the widespread adoption of agile methodologies emphasizes the need for educational approaches that prepare students for effective teamwork. Capstone courses in agile software engineering provide a valuable setting for students to experience collaborative work. However, the dynamics within these teams, particularly regarding gender, require further exploration. Objectives: This study examines teamwork experiences in an agile capstone course focusing on how team gender composition affects role distribution, teamwork quality, and gender bias. Methods: This study involved 94 teams comprising 561 students over two years (213 women and 348 men). We employed a mixed-methods approach, using four surveys (806 responses) and 12 in-depth interviews to capture ambition level, satisfaction with agile practices, and gender differences. Results: Team gender composition significantly impacted teamwork quality (TWQ), with female-dominated but gender-diverse teams showing the highest TWQ scores. Interestingly, women in token positions in male-majority teams reported unexpectedly high satisfaction levels, challenging aspects of Kanter's tokenism theory. However, women were more likely to report that team members underestimated their competence, particularly in male-majority teams. Observed gender differences included a tendency for women to engage more in design and process-oriented tasks, while men predominantly focused on programming and technical aspects such as architecture and algorithms. Conclusion: This study underscores the importance of considering team composition in relation to gender in agile teams, as we found this aspect greatly impacts team dynamics. Further, the implementation of agile practices in capstone courses yielded high overall TWQ scores, indicating significant educational value. Our findings provide insights for both software engineering education and agile practitioners. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
66	Learning to Work From Home: How Novice and Experienced Software Professionals Compare Online and In-person Collaboration	Industry-sponsored capstone experiences that involve developing software solutions create valuable opportunities for students to engage in teamwork and learn important career skills just prior to joining the software workforce. The collaboration and communication in capstone projects, which typically happen in person, were forced to move online during the COVID-19 pandemic. We leveraged the transition to conduct a two-stage mixed-methods study that compared the in-person and online capstone experiences of students (novice software developers) and project sponsors (experienced industry professionals) to understand how novices engage in online collaboration. We uncovered that novices find online collaboration more challenging compared to experienced professionals. The challenges faced by the novices and the misalignment with experienced professionals point to a number of avenues for training novice software professionals on effective engagement in geographically distributed collaboration, which is increasingly the norm in the software industry. © 2025 Elsevier B.V., All rights reserved.	☑	☐	☐	☐	☐	☐	☐	☑
67	Impact of software development processes on the outcomes of student computing projects: A tale of two universities	Context: Project-based courses are more and more commonly used as an opportunity to teach students structured methods of developing software. Two well-known approaches in this area – traditional and Agile – have been successfully applied to drive academic projects. However too often the default is still to have no organizational process at all. While a large variety of software development life-cycle models exists, little guidance is available on which one to choose to fit the context of working with students. Objective: This paper assesses the impact of iterative, sequential and "hands-off" development approaches on the success of student computing projects. A structured, metric-based assessment scheme was applied to investigate team productivity, teamwork and the quality of the final product. Method: Empirical evidence was collected during a controlled experiment carried out at two engineering schools in Europe. More than 100 students at Bachelor's and Master's levels participated in the research, with varied software development and teamwork skill sets. Results: Similar patterns were observed among both sets of subjects, with iterative teams demonstrating the highest productivity and superior team cohesion but a decline in the quality of the final product. Sequential development led to a considerable improvement in the external quality characteristics of the software produced, owing to the method's stress on design activities. Conclusion: The findings of this study will be of use to educators interested in applying software development processes to student groupwork. A set of guidelines is provided for applying a structured way of working in a project-based course. © 2021 Elsevier B.V., All rights reserved.	☐	☐	☐	☐	☐	☐	☐	☐
SCIENCE DIRECT									
68	Design Sprint: Enhancing STEAM and engineering education through agile prototyping and testing ideas	Creating project-based learning experiences in the classroom where students learn in a team to solve complex problems and to develop creative and critical thinking is a challenge. Design Sprint (DS) is an agile methodology (implemented in 5 days) with the goal of creating innovative design based on user needs (User Experience). The objective of this work was to develop an Engineering Drawing classroom experience linked to the context of the current COVID-19 pandemic with the Design Sprint methodology. The experience had to involve the integration of theory and practice, the application of knowledge, the development of both hard and soft skills, and the empowerment of students to conduct research. 56 first-year students following three STEAM degrees at the University of A Coruña participated in this experience. The activities were designed for both face-to-face and remote learning. Microsoft Teams and Moodle were used for tutoring and for monitoring student progress. The Moodle Workshop tool was used for the evaluation of the prototypes that were developed and the projects were evaluated by video. The students defended their projects through a presentation in lightning talk format (Ignite). Evaluation rubrics were used following a triple approach: co-evaluation, hetero-evaluation and self-evaluation. The 3D design of the projects was developed with Autodesk software. A total of 18 projects were developed. Once the projects were completed, a survey was administered to evaluate the levels of student satisfaction. The survey results were very positive. The Design Sprint projects also showed positive effects on grades. The Design Sprint method has promoted an interactive learning environment. In addition to its simplicity, a further advantage of DS method is that all student dedication is planned. Students were therefore less likely to feel overloaded, all of which helps with better time management. The DS methodology is multipurpose, so it can be applied to various fields and subjects.	☐	☐	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
69	Robotic-based invESTigation and mOnitoring of Ross sEa with the PROTEUS unmanned marine vehicle - The RESTORE project	In the framework of the Programma Nazionale di Ricerche in Antartide (PNRA) Robotic-based invESTigation and mOnitoring of Ross sEa (RESTORE) project, during the XXXVIII Italian Expedition in Antarctica (2022–2023), the Portable RObotic TEchnology for Underwater Surveys (PROTEUS) Unmanned Marine Vehicle (UMV) was used to carry out multidisciplinary scientific surveys in the Ross Sea. (PROTEUS) is an innovative UMV, developed by the Marine Robotics research group of CNR-INM, which is particularly suitable for operating in extreme environments such as the polar ones, thanks to its compact size, lightweight design, modularity, reconfigurability, and open hardware and software architectures. In the first period of the Italian Antarctic scientific expedition (PROTEUS) was used underwater, in the Remotely Operated Vehicle (ROV) configuration. In a second phase it was transformed into a Unmanned Semi-Submersible Vehicle (USSV) for operating semi-submersed, and in a third and final phase it was further transformed into a Unmanned Surface Vehicle (USV) vehicle for operating on the sea surface. The versatility of Portable RObotic TEchnology for Underwater Surveys (PROTEUS) allowed the onboard installation and integration of several sensors to acquire a comprehensive collection of bio-geo-chemical and physical parameters of the water column (conductivity, temperature, pressure, dissolved oxygen, turbidity and chlorophyll), and acoustic and video data of the ice and the seabed. This paper presents a detailed description of the design of PROTEUS, encompassing its modular architecture, propulsion and power supply systems, navigation, guidance and control system, and sensory payload. The field campaign described herein will demonstrate the exceptional adaptability of PROTEUS which enables novel methodologies for oceanic observations and data collection. PROTEUS is a non-conventional marine vehicle not only because of its distinctive design, but also due to its hybrid nature, which allows it to operate as four different marine robotic vehicles: ROV, AUV, USSV and USV. Moreover, the unique mechanics, electronics, and software architecture enable all robotic subsystems to work in concert, ensuring the coordinated functioning of the entire mechatronic device. Ultimately, PROTEUS incorporates an integrated data processing system capable of collecting and combining environmental and robotic telemetry. This renders PROTEUS a remarkable example of an innovative ocean engineering platform dedicated to data campaigns in Antarctica.	☑	☐	☐	☐	☐	☐	☑	☐
70	Experiencing Education 5.0 for Civil Engineering	Nowadays, the construction industry's digital transformation processes are challenging civil engineering education. In general, students and society underestimate the civil engineering field as an outdated and archaic knowledge area. Despite the defragmentation and heterogeneity of the sector, advances have been made on the path for Construction 4.0 and 5.0 scenarios. From this perspective, the CONSTRUCT-Gequaltec group of the Faculty of Engineering of the University of Porto have been sponsoring research and technological development using hardware and software targeting construction management solutions. This paper presents the work in progress mainly connected to the curricular units regarding project management in the Construction Section of the Civil Engineering Department. It also targets assumptions of some expected results across the research and innovation priorities from ECTP Innovative Built Environment and United Nations Sustainable Development Goals (SDGs). Two fundamental elements of the teaching practices are presented as the Construction Sites of the Future (Laboratory-Based Education initiative) and Digital products catalogue targeting Digital Twins in Construction (Project Based Learning initiative). The learning outcomes address increasing knowledge of innovative solutions, delivering skills to deploy 5.0 actions, and targeting human-centred and sustainable attitudes and values.	☑	☐	☐	☐	☐	☐	☑	☐
71	Efficient Git Repository Analysis: A Modular Plugin Approach	Insight Git is an adaptable and extensible Python package that presents a comprehensive analysis of Git repositories through an interactive graphical user interface. Based on a modular architecture and plugin support, Insight Git adapts to the requirements that different developers, project managers, and educators handle in their daily development or evaluation activities. This work documents the development and implementation of Insight Git, demonstrating its potential as an effective tool for development teams as well as for computer science and software engineering educators.	☑	☐	☐	☐	☐	☐	☐	☑
72	Status indicators in software engineering group projects	A segment of studies on group structure and performance in software engineering (SE) project-based learning (PjBL) have focused on roles, including studies that use Belbin team roles and studies that address problematic roles such as social loafing. The present study focuses on the status, which is basically missing in SE PjBL studies, although relating to roles. The study investigates the aspects that students identified as indicators of rising or declining status in their project groups. The status theory was utilized as the framework that motivated the research and on which the results were reflected. An inductive qualitative content analysis was applied to learning reports in which students reflected on their statuses. The indicators of rising status included technical know-how, commitment, management responsibility, and idea ownership, while also group-level attributes such as a caring atmosphere and joint responsibility. The indicators of a declining status included aspects that appear as counterparts of rising status indicators, while also more refined aspects such as no one willing to be a leader or study background. The results are concluded to provide material for educating students about intra-group relations and promoting self-regulation for fruitful collaboration in groups. The authors believe that the results also initiate further PjBL research in which status theory can be utilized.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
73	A systematic mapping study on group work research in computing education projects	Context: For developing students' group- and teamwork skills needed in the team-oriented work environments of the software industry, the role of project-based learning is considered central. Yet there does not appear to be a proper mapping of the current group work research in the computer science project education literature. Thus, the current state of group work research in the research area is somewhat unknown. Objective: This study aims to form an overview of how research has addressed students' group work in the field of computer science (CS) and software engineering (SE) to identify research gaps as well as suitable topics for more detailed literature reviews. Methods: A systematic mapping study was used to investigate how group work in tertiary education has been undertaken in the literature during the past decade. Results: Based on the selected papers, the most investigated group work areas were related to the assessment of groups, group formation, communication, and cooperation. The research appeared to be quite narrowly focused on a few areas. Most of the papers were experience or evaluation research. A case study using interviews or questionnaires to gather data from a single course was the most representative type of study. The papers were mainly published in scientific conferences. The use of theoretical frameworks was limited, with a focus on a few established frameworks. Tuckman's group development theory was the predominant framework, while other commonly used concepts and theories include social loafing, Kolb's learning style theory, and the Big Five personality traits model. Conclusion: Out of 7515 papers screened, 225 were deemed eligible and analyzed. We conclude a need for more focused group work research in CS/SE student projects, in which education is inspected from particular perspectives. This would create identifiable lines of research and structure the research area. Relatedly, we suggest that the underused theoretical frameworks can inspire important research: group interventions would benefit from a socially shared regulated learning perspective, explicit use of justice theories would improve theoretical understandings of group behavior, and transactional distance would help analyze how students adopt a software process. Moreover, the research area could be precipitated by novel theoretical perspectives. For practitioners, those implementing a group project course can benefit from a large amount of literature on assessment and group formation, which are issues on which the teacher must take a position. We also include a lessons-learned summary for teachers. Generally, the present results outlining the field in a structured way can facilitate research-based teaching.	☑	☑	☐	☐	☐	☐	☐	☐
74	An Empirical Investigation Into the Influence of Software Communities' Cultural and Geographical Dispersion on Productivity	Estimating and understanding software development productivity represent crucial tasks for researchers and practitioners. Although different works focused on evaluating the impact of human factors on productivity, a few explored the influence of cultural/geographical diversity in software development communities. More particularly, all previous treatise addresses cultural aspects as abstract concepts without providing a quantitative representation. Improved knowledge of these matters might help project managers to assemble more productive teams and tool vendors to design software analytics toolkits that may better estimate productivity. This paper has the goal of enlarging the existing body of knowledge on the factors affecting productivity by focusing on cultural and geographical dispersion of a development community—namely, how diverse a community is in terms of cultural attitudes and geographical collocation of the members who belong to it. To reach this goal, we performed a mixed-method empirical study. First, we built a statistical model relating dispersion metrics with the productivity of 25 open-source communities on Github. Then, we performed a confirmatory survey with 140 practitioners. The key results of our study indicate that cultural and geographical dispersion considerably impact productivity, thus encouraging managers and practitioners to consider such aspects during all the phases of the software development lifecycle. We conclude our paper by elaborating on the main insights from our analyses and instilling implications that may drive further research.	☑	☐	☐	☐	☐	☐	☑	☐
75	Relating team atmosphere and group dynamics to student software development teams' performance	Context While the software engineering community (i.e., those involved with engineering software) is constantly in search of insights into team atmosphere and group dynamics and the way these issues impact team performance, little opportunities typically exist to explore this issue. Student projects offer an opportunity for us to understand these issues, and particularly if these students are on the verge of leaving university for post-study work and using similar practices to those used in industry. Objective We explore a range of student software development projects' data and students' open-ended responses to five group dynamics categories: communication, time management, commitment, problem analysis and solving, and initiative and involvement. Method We analyse both quantitative and qualitative data to study the variation in group dynamics across teams developing different software and how these variations correlated with team satisfaction. We also explore the group dynamics themes that evolve from students' open responses in relation to the five categories. Furthermore, we relate the prevalence of the themes to various software development performance metrics, before exploring the opportunity of predicting an optimum team dynamics. Results We observe variations in the way different teams work, but higher performing teams also committed more to their projects. Various group dynamics themes were evident among functional teams, and specific patterns were more pronounced when teams were productive. Further, while there is no specific group dynamics pattern that predicts project success, successful teams were most organised and reflective. Conclusion Competence may set the tone for positive group dynamics and team performance. Also, an achievement-driven orientation is as important as the soft skills and interpersonal aspects.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
76	Software engineering team project courses with industrial customers: Students' insights on challenges and lessons learned	Team project courses in software engineering allow students to apply their acquired disciplinary knowledge while developing essential skills needed to work in the software industry. This paper examines the challenges and lessons learned by students in two team project courses involving industrial customers. The first course involves small teams and less complex project, whereas the second course, has larger teams and more complex projects. Using thematic analysis, we analyzed 158 reports submitted by two cohorts of students across two successive team project courses. As per our findings most challenges and lessons learned pertain to soft skills, such as teamwork, working in remote and hybrid setting, and collaboration with industrial customers. The results show that challenges and lessons learned evolve as students progress to the second team project course, for example, managing changes and addressing individual skill gaps were more pronounced in the first project course, while students reported greater coordination, communication, and contribution issues in the second team project course. The alignment between the challenges faced and the lessons learned suggests that addressing challenges in teamwork, collaborating with industrial customers, and working in hybrid or remote settings helped students develop effective strategies to mitigate these challenges. This process offers a valuable learning experience for the students, enriching their professional growth.	☑	☑	☐	☐	☐	☐	☐	☐
77	The nexus of project management approaches in sustainable development: innovative behaviors as a mechanism in the Polish financial industry	Purpose Based on the social exchange theory, the aim of the present study is to examine the effects, both direct and indirect (through sustainability-oriented innovative behaviors [SIBs]), of sustainable project leadership (SPL) on sustainable project performance (SPP). Project management approaches (PMAs) (traditional, hybrid and agile) were examined as conditional factors in the "SPL–SIBs" relationship. Design/methodology/approach The study employs structural equation modeling based on data collected from 197 software engineering project team members working in the financial industry in Poland. Findings The study revealed that SPL significantly, positively affected SPP. It also provided evidence for the significant mediating impact of SIBs in the relationship between SPL and SPP and the conditional effect of agile and hybrid PMAs on the "SPL–SIBs" relationship. Originality/value The novelty of this work lies in introducing sustainable leadership into project management research, proposing and testing a unique and complex research framework, designing valid scales for measuring SPL and SPP, and suggesting many theoretical and empirical implications.	☑	☐	☐	☐	☐	☐	☑	☐
78	Simulating a sustainable biorefinery process as autonomous learning tool for collaborative work in the chemical engineering degree	To improve the skills of the students in the Sustainable Refinery and Biorefinery subject of the Degree in Chemical Engineering, a learning tool has been implemented based on collaborative work, autonomy, and the use of information and communication technologies (ICT). This project includes the development of simulations that allow students to apply the learned concepts in a practical way, which were then assessed with the help of a specific rubric created for this purpose. Once the instrument was designed and implemented, its evaluation was carried out using a specific questionnaire. The results indicated that the use of simulation software in the classroom (with a 95% acceptance rate among students) had a positive impact on the students, facilitating the comprehension of the content (scored a 3.9 out of 1–5 Likert scale), supporting their academic performance (close to 10% average increase in the student qualifications), and fostering their professional profile in the field of Chemical Engineering (as supported by the rating of 4 out of 1–5 Likert scale). In addition, the constructive feedback provided by the students enabled the teaching team to identify areas for improvement and to consider future changes and updates to the pedagogical approach (as the organization of the seminars). Therefore, the use of different chemical process simulation software packages resulted in an essential strategy to develop the autonomous learning of the students and constituted a successful methodological tool, also preparing them to better meet the challenges of the industrial sector.	☑	☐	☐	☐	☐	☐	☑	☐
79	GitHub marketplace for automation and innovation in software production	Context: GitHub, renowned for facilitating collaborative code version control and software production in software teams, expanded its services in 2017 by introducing GitHub Marketplace. This online platform hosts automation tools to assist developers with the production of their GitHub-hosted projects, and it has become a valuable source of information on the tools used in the Open Source Software (OSS) community. Objective: In this exploratory study, we introduce GitHub Marketplace as a software marketplace by exploring the Characteristics, Features, and Policies of the platform comprehensively, identifying common themes in production automation. Further, we explore popular tools among practitioners and researchers and highlight disparities in the approach to these tools between industry and academia. Method: We adopted the conceptual framework of software app stores from previous studies and used that to examine 8,318 automated production tools (440 Apps and 7,878 Actions) across 32 categories on GitHub Marketplace. We explored and described the policies of this marketplace as a unique platform where developers share production tools for the use of other developers. Furthermore, we conducted a systematic mapping of 515 research papers published from 2000 to 2021 and compared open-source academic production tools with those available in the marketplace. Results: We found that although some of the automation topics in literature are widely used in practice, they have yet to align with the state-of-practice for automated production. We discovered that practitioners often use automation tools for tasks like "Continuous Integration" and "Utilities", while researchers tend to focus more on "Code Quality" and "Testing". Conclusion: Our study illuminates the landscape of open-source tools for automation production. We also explored the disparities between industry trends and researchers' priorities. Recognizing these distinctions can empower researchers to build on existing work and guide practitioners in selecting tools that meet their specific needs. Bridging this gap between industry and academia helps with further innovation in the field and ensures that research remains pertinent to the evolving challenges in software production.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
80	Architecting a decision support system for continuing supervision of commercial in-space servicing	The rapid development of in-space servicing technology and other novel space capabilities requires robust and transparent governance frameworks to ensure long-term space sustainability and adherence to international regulations, notably Article VI of the Outer Space Treaty. Article VI requires that signatory states provide continuing supervision over non-governmental space activities, a mandate becoming increasingly more challenging to fulfill due to the accelerating pace of commercial space innovations. In previous work published by the authors, a Systems Architecture Framework analysis investigated the governance of in-space servicing in the U.S. and the corresponding Stakeholder Need misalignments with current authorization and supervision processes. The initial research provided insights into the apparent Need for a Decision Support System addressing the practical challenges faced in the operational supervision of in-space servicing activities. In response, this paper roadmaps the application of the Environment-Vulnerability-Decision-Technology (EVDT) systems engineering framework into the realm of space sustainability challenges, such as for authorization and supervision of commercial in-space servicing. Originally conceived by the Space Enabled research group at MIT's Media Lab, the EVDT framework has demonstrated its effectiveness in facilitating sustainable development decision-making through analysis of complex socio-environmental-technical systems across various terrestrial applications. Historical uses of EVDT span across aiding flood resilience in Indonesia, promoting mangrove preservation in Brazil, managing invasive plant species in Benin, revitalizing cranberry wetlands in the U.S., analyzing environmental injustice in prison landscapes, and urban planning strategies during the pandemic. Most recently, the inaugural adaptation of the EVDT framework to the space domain shows potential to enhance collision avoidance operation decisions for a Stakeholder within NASA. This paper proposes the expansion of the EVDT framework to broader space sustainability challenges, focusing on continuing supervision as the primary use-case, where this prototype's capability to model and analyze hypothetical commercial in-space satellite servicing missions under U.S. jurisdiction will demonstrate the potential of EVDT to enhance space situational awareness (SSA) and space domain awareness. These operations are critical for collision prediction and consequence, risk assessment, and the implementation of sustainable operational practices. We introduce the plan for developing the Continuing Supervision EVDT software prototype, using a MATLAB-based method characterized by a modular architecture to facilitate integration and extension of functionality. The paper also introduces terminology, key concepts, objectives, and the use of the Systems Architecture Framework method within the EVDT software development process. The software design enables Stakeholders to custom-build and adapt their models to different space sustainability scenarios, improving code reuse, reducing development time, and simplifying interactions for external users and future space-based EVDT projects. The implementation of this Decision Support System has the potential to influence the authorization and supervision of novel space missions and the evolution of supporting SSA technologies, ultimately contributing to the responsible and sustainable use of the space environment. It helps ensure compliance with international space laws and promotes sustainability by equipping Stakeholders with software toolsets capable of simulating the orbital dynamics of spacecraft through mission phases. The paper also envisions the extensive application of the EVDT framework to an array of other space sustainability challenges, such as environmental sensitivity, debris mitigation, resource utilization, and planetary protection. Ultimately, the expansion of the EVDT framework into the domain of space sustainability will empower policymakers, commercial space operators, and other Stakeholders with an adaptive simulation tool that not only conforms to the current space governance systems but also flexibly shapes to future space policies, encouraging responsible stewardship over the space environment.	☑	☐	☐	☐	☐	☐	☑	☐
81	Collaborative Online International Learning (COIL) in chemical engineering: Preparing students for multicultural and international work environments	In today's interconnected society, chemical engineering students must be prepared to work in international and multicultural environments. However, in our experience, current chemical engineering curricula often fail to develop these competencies. This study aims to demonstrate the benefits of Collaborative Online International Learning (COIL) in chemical engineering education. For the first time, the COIL approach has been implemented in a simulation course. In addition to preparing students for international and multicultural work environments, this experience enhances their problem-solving and critical-thinking skills. Unlike other COIL applications, this project allows for multiple valid solutions, though not all are necessarily optimal. After two successful COIL projects involving chemical engineering students from the Universidad de Valladolid (Spain), the Universidad Nacional de Colombia, and the Universidad Autónoma de Nuevo León (Mexico), students reported feeling more confident in their knowledge and abilities, better prepared for multicultural and international work environments, and more capable of performing well in their first job. In both project editions, survey responses to related questions averaged above 4 out of 5. Key takeaways from this work are that, to accomplish the objectives of a COIL, it is essential to define the project timeline in advance, ensure a similar level of knowledge among students, confirm software access, establish a unified communication platform, and conduct individual kickoff meetings for each team. Additionally, effective international collaboration is more likely when no more than 50 % of a team's members come from the same institution.	☑	☐	☐	☐	☐	☐	☐	☑
82	General Variable Neighborhood Search for the optimization of software quality	In the area of Search-Based Software Engineering, software engineering issues are formulated and tackled as optimization problems. Among the problems within this area, the Software Module Clustering Problem (SMCP) consists of finding an organization of a software project that minimizes coupling and maximizes cohesion. Since modular code is easier to understand, the objective of this problem is to increase the quality of software projects, thus increasing their maintainability and reducing the associated costs. In this work we study a recently proposed objective function named Function of Complexity Balance (FCB). Since this problem has been demonstrated to be NP-hard, we propose a new heuristic algorithm based on the General Variable Neighborhood Search (GVNS) schema to tackle the problem. For the GVNS, we propose six different neighborhood structures and categorize them into three different groups. Then, we analyze their contribution to the results obtained by the algorithm. In order to improve the efficiency of the proposed approach, we leverage domain-specific information to perform incremental evaluations of the objective function and to explore only areas of interest in the search space. The proposed algorithm has been tested over a set of real world software repositories, achieving better results than the previous state-of-the-art method, a Hybrid Genetic Algorithm, in terms of both quality and computing times. Furthermore, the relevance of the improvement produced by our proposal has been corroborated by non-parametric statistical tests.	☑	☐	☐	☐	☐	☐	☐	☑
83	Loss Assessment of Non-Ductile RC Buildings Accounting for Bidirectional Ground Motion	The ongoing seismic events in the world highlight the significant vulnerability of the existing building stock. In particular, in Italy, several structures have been designed without seismic criteria and showed significant damage after recent earthquakes, causing many casualties and great economic losses. Earthquake impact assessment has become one of the most investigated topics by several research groups and within national and international projects. In Italy, for example, the National Department of Civil Protection has allocated several economic funds within the ReLUIS project to define methodologies able to support the decision-making process for seismic prevention activities and risk reduction. This work aims to assess economic losses using the methodology proposed by FEMA-P58, according to the Performance-Based Earthquake Engineering method, evaluating the influence of bi-directional ground motion on loss estimates. The analyzed case study is a three-dimensional nonlinear frame characterized by one weak and one strong seismic directions and modeled using OpenSees software. The model considers the joint deformability and the flexural and shear damaging behavior of beams and columns. Thirty time-history analyses were performed considering bi-directional ground motions applied in the two main directions of the building and applying the higher spectral acceleration firstly in the weaker direction and then in the stronger one. Eight increasing hazard levels were selected for the nonlinear time-history analyses, and the corresponding structural analysis results, in terms of interstory drift ratio and peak floor acceleration, were implemented for the loss assessment. The results of the loss assessment are then discussed in terms of expected annual loss and loss hazard curve.	☑	☐	☐	☐	☐	☐	☑	☐
IEE XPLORE										

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
84	With Great Humor Comes Great Developer Engagement	The worldwide collaborative effort for the creation of software is technically and socially demanding. The more engaged developers are, the more value they impart to the software they create. Engaged developers, such as Margaret Hamilton programming Apollo 11, can succeed in tackling the most difficult engineering tasks. In this paper, we dive deep into an original vector of engagement – humor – and study how it fuels developer engagement. First, we collect qualitative and quantitative data about the humorous elements present within three significant, real-world software projects: faker, which helps developers introduce humor within their tests; lolcommits, which captures a photograph after each contribution made by a developer; and volkswagen, an exercise in satire, which accidentally led to the invention of an impactful software tool. Second, through a developer survey, we receive unique insights from 125 developers, who share their real-life experiences with humor in software.Our analysis of the three case studies highlights the prevalence of humor in software, and unveils the worldwide community of developers who are enthusiastic about both software and humor. We also learn about the caveats of humor in software through the valuable insights shared by our survey respondents. We report clear evidence that, when practiced responsibly, humor increases developer engagement and supports them in addressing hard engineering and cognitive tasks. The most actionable highlight of our work is that software tests and documentation are the best locations in code to practice humor.CCS CONCEPTS• Software and its engineering → Programming teams; • Social and professional topics → Cultural characteristics.LAY ABSTRACTModern software applications are built piecemeal. They are composed of a large pool of contributions made by individual software developers. These developers spend long hours brainstorming with colleagues, solving challenging engineering problems, and writing code. Software companies put different strategies in place to engage their developers and support them in coping with these hard tasks. In particular, some organizations and groups of developers decide to nurture the creative and expressive aspects of software development. In this work, we study a universal human trait that has been key for creativity since the dawn of civilization: humor. Humorous software code is a powerful deterrent to boredom. It can uplift the mood of the developer who writes it, as well as a peer, separated by time and space, who chances upon this nugget in the codebase. For example, Leia Organa is as perfect as John Doe for a placeholder customer name while testing an e-commerce application. A gallery of photographs can serve as a cheerful journal of the Wednesdays spent hacking away with a colleague. Software may also act as a humorous, satirical piece on socio-political issues. In this paper, we discover the different ways in which software can be a source of developer amusement and engagement, and help foster a sense of community among them.	☑	☐	☐	☐	☐	☐	☐	☑
85	What Makes Effective Leadership in Agile Software Development Teams?	Effective leadership is one of the key drivers of business and project success, and one of the most active areas of management research. But how does leadership work in agile software development, which emphasizes self-management and self-organization and marginalizes traditional leadership roles? To find out, this study examines agile leadership from the perspective of thirteen professionals who identify as agile leaders, in different roles, at ten different software development companies of varying sizes. Data from semi-structured interviews reveals that leadership: (1) is dynamically shared among team members; (2) engenders a sense of belonging to the team; and (3) involves balancing competing organizational cultures (e.g. balancing the new agile culture with the old milestone-driven culture). In other words, agile leadership is a property of a team, not a role, and effectiveness depends on agile team members' identifying with the team, accepting responsibility, and being sensitive to cultural conflict.	☑	☐	☐	☐	☐	☐	☑	☐
86	What Leads to a Confirmatory or Disconfirmatory Behavior of Software Testers?	Background: The existing literature in software engineering reports adverse effects of confirmation bias on software testing. Confirmation bias among software testers leads to confirmatory behavior, which is designing or executing relatively more specification consistent test cases (confirmatory behavior) than specification inconsistent test cases (disconfirmatory behavior). Objective: We aim to explore the antecedents to confirmatory and disconfirmatory behavior of software testers. Furthermore, we aim to understand why and how those antecedents lead to (dis)confirmatory behavior. Method: We follow grounded theory method for the analyses of the data collected through semi-structured interviews with twelve software testers. Results: We identified twenty antecedents to (dis)confirmatory behavior, and classified them in nine categories. Experience and Time are the two major categories. Experience is a disconfirmatory category, which also determines which behavior (confirmatory or disconfirmatory) occurs first among software testers, as an effect of other antecedents. Time Pressure is a confirmatory antecedent of the Time category. It also contributes to the confirmatory effects of antecedents of other categories. Conclusion: The disconfirmatory antecedents, especially that belong to the testing process, e.g., test suite reviews by project team members, may help circumvent the deleterious effects of confirmation bias in software testing. If a team's resources permit, the designing and execution of a test suite could be divided among the test team members, as different perspectives of testers may help to detect more errors. The results of our study are based on a single context where dedicated testing teams focus on higher levels of testing. The study's scope does not account for the testing performed by developers. Future work includes exploring other contexts to extend our results.	☑	☐	☐	☐	☐	☐	☐	☑
87	Weighted Multi-Skill Resource Constrained Project Scheduling: A Greedy and Parallel Scheduling Approach	This study addresses the Weighted Multi-Skill Resource Constrained Project Scheduling Problem (W-MSRCSPSP) with the aim of minimizing software project makespan. Unlike previous works, our investigation regards heterogeneous resources characterized by varying skill proficiency levels. Another major problem with existing methodologies is the potential underutilization of human resources due to varying task durations. This work introduces an innovative scheduling approach known as the Greedy and Parallel Scheduling (GPS) algorithm to handle the said issues. GPS focuses on assigning the most suitable resources available to project activities at each scheduling point. The fundamental goal of our proposed approach is to reduce resource wastage while efficiently allocating surplus resources, if any, to project tasks, ultimately leading to a decrease in the makespan. To empirically evaluate the efficacy of the GPS algorithm, we conduct a comparative analysis against the Parallel Scheduling Scheme (PSS). The advantage of our proposed approach lies in its ability to optimize the utilization of available resources, resulting in accelerated project completion. Results from extensive simulations substantiate this claim, demonstrating that the GPS scheme outperforms the PSS approach in minimizing project duration.	☑	☐	☐	☐	☐	☐	☐	☑
88	Using PBL and Agile to Teach Artificial Intelligence to Undergraduate Computing Students	Project-based learning (PBL) is an active learning methodology focused on developing both soft and hard skills by solving real-world problems. In PBL, teachers act as facilitators while students take charge of their own learning. While the practice of this methodology in computing education has been growing in recent years, it still poses some challenges that need to be addressed. Integrating Scrum and Agile methodologies into PBL can be a valuable addition when teaching computing subjects to students. Therefore, this paper presents a case study of a successful implementation of PBL with Scrum, applying Agile values and principles to teach Artificial Intelligence to undergraduate students. This study contributes to the limited research on Scrum in education, as well as helps bridge the research gap in AI teaching and learning. The case study involved 30 students from an undergraduate computing program, divided into five groups, who successfully developed five different Machine Learning (ML) models to tackle the challenging real-life problem of breast cancer prediction for the Cancer Institute of the State of São Paulo. The findings of the study indicate that the students effectively utilized Scrum and Agile methodologies throughout the process and expressed satisfaction with the approach. Additionally, they developed problem-solving abilities, critical thinking and communication skills, teamwork capabilities, and gained experience in working with real-life situations and problems. The study also demonstrates that the proposed technique aligns with the foundations of the PBL approach in computing education discussed in previous literature. It serves as a valuable resource for future research on PBL implementation, by comparing similarities and differences with existing literature and discussing the strategies employed to address implementation challenges.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
89	Understanding the Career Mobility of Blind and Low Vision Software Professionals	Context: Scholars in the software engineering (SE) research community have investigated career advancement in the software industry. Research topics have included how individual and external factors can impact career mobility of software professionals, and how gender affects career advancement. However, the community has yet to look at career mobility from the lens of accessibility. Specifically, there is a pressing need to illuminate the factors that hinder the career mobility of blind and low vision software professionals (BLVSPs). Objective: This study aims to understand aspects of the workplace that impact career mobility for BLVSPs. Methods: We interviewed 26 BLVSPs with different roles, years of experience, and industry sectors. Thematic analysis was used to identify common factors related to career mobility. Results: We found four factors that impacted the career mobility of BLVSPs: (1) technical challenges, (2) colleagues' perceptions of BLVSPs, (3) BLVSPs' own perceptions on managerial progression, and (4) BLVSPs' investment in accessibility at the workplace. Conclusion: We suggest implications for tool designers, organizations, and researchers towards fostering more accessible workplaces to support the career mobility of BLVSPs. CCS CONCEPTS • Human-centered computing → Empirical studies in collaborative and social computing; Empirical studies in accessibility.	☑	☐	☐	☐	☐	☐	☑	☐
90	Uncovering Determinants of Code Quality in Education via Static Code Analysis	The role of static code analysis in enhancing the quality of software codes is widely acknowledged. Static code analysis facilitates the examination of code for irregularities without program execution, which significantly impacts project quality. Furthermore, tools for static code analysis serve as educational aids, imparting essential lessons on coding practices. Motivated by the growing complexity of software projects and the pivotal role of code quality in academic performance within computing disciplines, this research examines over 500 student projects using static code analysis tools. The aim is to determine metrics that influence the code quality of student projects. The study investigates how metrics, such as project setup, influence code quality and students' academic performances. By adopting a broad approach, the investigation determines the overall impact of these metrics on the technical integrity of software engineering projects and academic outcomes. Insights derived from this study are anticipated to enhance teaching strategies and curriculum development, aiming to improve academic performance by promoting better code quality.	☑	☐	☐	☐	☐	☐	☐	☑
91	Transmit Diversity-Achieving Quadrature Index Modulation Aided Media-Based Modulation	In this paper, a new schematic of quadrature index modulation (QIM) with generalized spatial constellations is proposed and amalgamated with the media-based modulation (MBM) system for not only achieving the transmit diversity gains but also increasing the spectral efficiency (SE). Based on this, this new scheme may be named as Transmit Diversity-achieving QIM-MBM (TD-QIM-MBM). Specifically, all available transmit antennas (TAs), whose each TA is equipped with $n_{\mathrm {rf}}$ radio frequency (RF) mirrors, are firstly grouped into G groups of TAs, each group of which is assigned to one corresponding Bits-Modulator. Then, one mapped three dimension (3D) signal constellation point (CP) is copied into G versions and then are respectively fed into G groups of Bits-Modulators. In each group of Bits-Modulator, with the aid of the antenna index (AI) bits of three spatial domains, one version of 3D signal CP is modulated into one QIM vector symbol. Furthermore, the in-phase and quadrature vectors of the QIM vector symbol are respectively modulated on the specified channel states with the in-phase and quadrature channel index (CI) bits and then formed into one MBM vector symbol. On the design of G groups of Bits-Modulators, G MBM vector symbols are obtained and concatenated into one transmitted MBM vector (TMV) symbol. Finally, in order to outstand the advantage of the proposed TD-QIM-MBM, the SEs and the squared minimum Euclidean distances (MEDs) of the TD-QIM-MBM are provided in comparisons with other schemes, and the average bit error probability is also provided. In simulation results with the Monte Carlo method, it demonstrates that the TD-QIM-MBM outperforms the classic design schemes (e.g., GSM-MBM and QSM-MBM) in terms of the bit error rate (BER) performance for enhancing the reliability of MBM-based wireless communications.	☑	☐	☐	☐	☐	☐	☑	☐
92	Towards Trustworthy AI Engineering - A Case Study on integrating an AI audit catalog into MLOps processes	In recent years, Machine Learning Operations (MLOps) has become increasingly important as more and more Machine Learning (ML) based applications are brought into production. With this widespread, attention must be paid to the application's trustworthiness. Numerous methods and tools have already been developed in the area of trustworthy AI. However, the integration of those into the MLOps cycle and in particular into the pipeline engineering process is missing. To address this open problem, we analysed an AI audit catalog and translated the respective requirements into a healthcare IT service provider's MLOps process. In this work, we describe the translation process and present the insights obtained via a case study. Our work highlights the necessary considerations for professionals and the scientific community when dealing with similar challenges in Trustworthy AI engineering and operations and provides clear recommendations.CCS CONCEPTS• Software and its engineering → Agile software development.	☑	☐	☐	☐	☐	☐	☑	☐
93	Towards Implementing and Evaluating AI-Assisted Pull Requests in Software Engineering Education	Pull requests allow developers to suggest and review codebase changes collaboratively. This process is standard for maintaining code quality and following best practices. The recent emergence of Large Language Models like ChatGPT and GitHub Copilot has shown great potential in improving coding efficiency and accuracy in software engineering. This paper outlines a study design to explore the integration of an AI tool to streamline PR reviews in a software engineering course. By incorporating the pr-agent into the curriculum, the study aims to evaluate its impact on students' coding skills, understanding of PR processes, and overall learning experience. The evaluation strategy includes collecting quantitative and qualitative feedback from students to assess the effectiveness of the tool. The results will offer insight into the feasibility and benefits of integrating AI tools in software engineering education.	☑	☐	☐	☐	☐	☐	☐	☑
94	"This Is Damn Slick!" Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors	Twitter is widely used by software developers. But how effective are tweets at promoting open source projects? How could one use Twitter to increase a project's popularity or attract new contributors? In this paper we report on a mixed-methods empirical study of 44,544 tweets containing links to 2,370 open-source GitHub repositories, looking for evidence of causal effects of these tweets on the projects attracting new GitHub stars and contributors, as well as characterizing the high-impact tweets, the people likely being attracted by them, and how they differ from contributors attracted otherwise. Among others, we find that tweets have a statistically significant and practically sizable effect on obtaining new stars and a small average effect on attracting new contributors. The popularity, content of the tweet, as well as the identity of tweet authors all affect the scale of the attraction effect. In addition, our qualitative analysis suggests that forming an active Twitter community for an open source project plays an important role in attracting new committers via tweets. We also report that developers who are new to GitHub or have a long history of Twitter usage but few tweets posted are most likely to be attracted as contributors to the repositories mentioned by tweets. Our work contributes to the literature on open source sustainability.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
95	The Influence of Software Engineering Education on Developing Teamwork Skills in Both Genders	Teamwork is recognized as a highly valuable skill for any software engineer. In computer science programs it is typically developed through project-based courses, though these courses often do not adopt a gender-oriented approach. Women are underrepresented in the software industry, and there is limited knowledge about their teamwork competencies. While some studies compare teamwork skills between men and women, few have explored the impact of education on the development of this skill, particularly in relation to different aspects of teamwork. By comparing teamwork quality between genders at different program levels, we intend to gain valuable insights about students' initial teamwork skills and the effect of software engineering education over time. To do this, we have carried out an empirical study of two project-based software engineering courses in our Computer Science and Engineering program: an introductory software engineering course and a capstone course. In each case we evaluate different aspects of teamwork by applying the same peer assessment instrument along 13 semesters. Our study revealed no significant gender-based disparities within the courses under study, indicating that such differences are absent both in students' initial abilities and final competencies. What we did observe, however, is that women's teamwork assessments are more homogeneous over time than the men's, where we see significant improvements in Attitude, Motivation, Coordination and Communication between courses. Our results lead us to believe that, it takes men longer to reach the same capacity for working in teams than women, but in the end, students of both genders reach similar teamwork capabilities in all teamwork quality dimensions. Therefore, women have just as much potential to excel as software engineers as men, especially when it comes to teamwork.	☑	☑	☐	☐	☐	☐	☐	☐
96	The Effects of Computational Resources on Flaky Tests	Flaky tests are tests that non-deterministically pass and fail in unchanged code. These tests can be detrimental to developers' productivity. Particularly when tests run in continuous integration environments, the tests may be competing for access to limited computational resources (CPUs, memory etc.), and we hypothesize that resource (un)-availability may be a significant factor in the failure rate of flaky tests. We present the first assessment of the impact that computational resources have on flaky tests, including a total of 52 projects written in Java, JavaScript and Python, and 27 different resource configurations. Using a rigorous statistical methodology, we determine which tests are RAFTs (Resource-Affected Flaky Tests). We find that 46.5% of the flaky tests in our dataset are RAFTs, indicating that a substantial proportion of flaky-test failures happen depending on the resources available when running tests. We report RAFTs and configurations to avoid them to developers, and received interest to either fix the RAFTs or to improve the specifications of the projects so that tests would be run only in configurations that are unlikely to encounter RAFT failures. Although most test suites in our dataset are executed quite quickly (under one minute) in a baseline configuration, our results highlight the possibility of using this methodology to detect RAFT to reduce the cost of cloud infrastructure for reliably running larger test suites.	☑	☐	☐	☐	☐	☐	☑	☐
97	The Design and Realization of a Self-Hosted and Open-Source Agile Internal Development Platform	Platform Engineering (PE) is a growing area of Software Engineering, with many facets, including the elusive concept of Internal Development Platforms (IDPs). They integrate various technologies and tools to support the realization of software projects, especially their managing and developing roles, fostering Agile collaboration and enhancing team productivity. In particular for the new trends of Agile remote working and for Agile applied to the Digital Transformation in the Public Sector. While proprietary IDPs abound, academic research that could naturally lead to open-source IDPs remain scarce. In particular for Scrum, a fundamental Agile methodology in the strategic area of Digital Transformation in the Public Sector, where IDP self-hosting is also appropriate. This paper addresses this gap by proposing the design and realization of the first, to the best of our knowledge, IDP supporting Scrum, being open-source and with self-hosting capabilities. The end-product of our research is the Compositional Agile System (CAS), an extendable microservice architecture-based IDP. Its design, as well as CAS, makes full use of hands-on previous successful experiences of IDP development by members of this team, specifically non-trivial tasks in Italian Public Administration and to foster remote working in University Software Engineering classes, a practice started during the pandemic but now a standard. We offer several architectural and example scenarios where CAS can be of use, concentrating on the Public Administration. Particularly relevant is the use of CAS in analyzing a pillar of Italian Digital Administration available to its citizens, i.e., Applo. The software is available as an open resource to the research community.	☑	☐	☐	☐	☐	☐	☑	☐
98	The Art and Practice of Data Science Pipelines: A Comprehensive Study of Data Science Pipelines In Theory, In-The-Small, and In-The-Large	Increasingly larger number of software systems today are including data science components for descriptive, predictive, and prescriptive analytics. The collection of data science stages from acquisition, to cleaning/curation, to modeling, and so on are referred to as data science pipelines. To facilitate research and practice on data science pipelines, it is essential to understand their nature. What are the typical stages of a data science pipeline? How are they connected? Do the pipelines differ in the theoretical representations and that in the practice? Today we do not fully understand these architectural characteristics of data science pipelines. In this work, we present a three-pronged comprehensive study to answer this for the state-of-the-art, data science in-the-small, and data science in-the-large, Our study analyzes three datasets: a collection of 71 proposals for data science pipelines and related concepts in theory, a collection of over 105 implementations of curated data science pipelines from Kaggle competitions to understand data science in-the-small, and a collection of 21 mature data science projects from GitHub to understand data science in-the-large. Our study has led to three representations of data science pipelines that capture the essence of our subjects in theory, in-the-small, and in-the-large.	☑	☐	☐	☐	☐	☐	☑	☐
99	Teaching Software Engineering: A Continuously Evolving Perspective	Software engineering is in a continuous process of change and evolution led by the dynamicity of our world, and hence, by the aim to adapt to and meet the current and future society's needs. From the educational point of view, one of the main challenges consists in what is meaningful and useful to teach in a software engineering course to enable students to have a solid background and to address and lead current and future software development. In this paper, we outline the importance of continuously evolving education (e.g., course material, and teaching methods) to keep up with the changes in software engineering. This enables students to learn the actual and relevant topics, skills, and technologies in a continuously evolving field where new methodologies, technologies, and best practices are constantly being added. We consider this a valuable development in software engineering education, as it allows for an active and relevant learning experience for students. The novelty of our paper consists of 1) a continuous evolution perspective on software engineering education, 2) exploiting various types of inputs driving evolution, and 3) a set of concrete examples and guidelines that can be generalized and applied to similar courses or in other fields.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
100	Teaching Software Development for Real-World Problems Using a Microservice-Based Collaborative Problem-Solving Approach	Experienced and skillful software developers are needed in organizations to develop software products effective for their business with shortened time-to-market. Such developers will not only need to code but also be able to work in teams and collaboratively solve real-world problems that organizations are facing. It is challenging for educators to nurture students to become such developers with strong technical, social, and cognitive skills. Towards addressing the challenge, this study presents a Collaborative Software Development Project Framework for a course that focuses on learning microservices architectures and developing a software application for a real-world business. Students get to work in teams to solve a real-world problem of their own choice. They are given opportunities to recognize that the software development process goes beyond writing code and that social and cognitive skills in engaging with each other are also essential. By adopting microservices architectures in the course, students learn to break down the functionalities of their applications into smaller pieces of code with standardized interfaces that can be developed, tested, and deployed independently. This not only helps students to learn various technical skills needed for developing and implementing the functionalities needed by the application in the form of microservices but also facilitates task allocation and coordination among their team members and provides a platform for them to solve problems collaboratively. Upon completion of their projects, students are also asked to reflect on their development process and encouraged to think beyond the basics for better software design and development approaches. The course curriculum incorporates the framework, especially for the student team projects. The earlier teaching weeks introduce a combination of concepts and lab exercises to students as the building blocks. The survey studies show that the framework is effective in enhancing the students' learning of technical, social, and cognitive skills, while further improvements, such as closer collaboration with other courses, can be done to improve a holistic learning curriculum.	☐	☐	☐	☐	☐	☐	☐	☐
101	Supporting Peripheral Perception in Distributed Teams by Enforced Exposure to Chat Messages	Members of colocated teams benefit from being able to peripherally perceive ongoing conversations separating the useful information from the rest of the ambient sound. Instead of oral communication, distributed teams usually rely on chat. We developed an approach to supporting peripheral perception in distributed teams by enforced exposure to chat messages and implemented it as a chat client for Slack. The idea is to expose the team members to receive messages in order to emulate natural peripheral perception of oral communication that can be observed in colocated teams. We assessed how well enforced exposure to chat messages supports peripheral perception by following communication intensity, message relevance, and distraction in two experiment settings: content-based and periodical message displaying. The experiments were performed with four student teams in a week to two weeks' time span. Seven team members had our chat client installed, while the rest of them (14) did not and used the unadapted chat clients they commonly use. The experiments were accompanied by a survey. Overall, enforced exposure to chat messages was perceived positively by the participants, especially in content-based message displaying. Communication intensity was clearly higher there, too. A strong correlation between the distraction caused by enforced exposure to chat messages and the communication intensity has been confirmed.	☑	☐	☐	☐	☐	☐	☐	☑
102	Studying the Influence and Distribution of the Human Effort in a Hybrid Fitness Function for Search-Based Model-Driven Engineering	Search-Based Software Engineering (SBSE) offers solutions that efficiently explore large complex problem spaces. To obtain more favorable solutions, human participation in the search process is needed. However, humans cannot handle the same number of solutions as an algorithm. We propose the first hybrid fitness function that combines human effort with human simulations. Human effort refers to human participation for providing evaluations of candidate solutions during the search process, whereas human simulations refer to recreations of a scenario in a specific situation for automatically obtaining the evaluation of candidate solutions. We also propose three variants for the hybrid fitness function that vary in the distribution of human effort in order to study whether the variants influence the performance in terms of solution quality. Specifically, we leverage our hybrid fitness function to locate bugs in software models for the video games of game software engineering. Video games are a fertile domain for these hybrid functions because simulated players are naturally developed as part of the video games (e.g., bots in First-Person Shooters). Our evaluation is at the scale of industrial settings with a commercial video game (Play Station 4 and Steam) and 29 professional video game developers. Hybridizing the fitness function outperforms the results of the best baseline by 33.46% in F-measure. A focus group confirms the acceptance of the hybrid fitness function. Hybridizing the fitness function significantly improves the bug localization process by reducing the amount of tedious manual work and by minimizing the number of bugs that go unnoticed. Furthermore, the variant that obtains the best results is a counter-intuitive result that was under the radar of the interactive SBSE community. These results can help not only video game developers to locate bugs, but they can also inspire SBSE researchers to bring hybrid fitness functions to other software engineering tasks.	☑	☐	☐	☐	☐	☐	☑	☐
103	Strategies for Dealing With Software Product Management Challenges	Software product management, which involves multiple processes and responsibilities and links to many activities within the organisation, increases the success rate of IT projects. However, the adoption of product management activities and the implementation of the software product manager role itself varies between companies. This research explores the topic of software product managers' strategies for dealing with challenges in their work. The aim of the study is to identify and evaluate solutions to the main problems that affect the software product management process and to provide guidelines for dealing with them. For this purpose, a focus group method was adopted and 47 software product managers participated in 15 focus groups. This was followed by a survey to assess the comprehensibility and effectiveness of the solutions identified. The paper proposes a list of 39 solutions to the 5 most common problems in the work of product managers. Furthermore, based on the identified solutions and the results of discussions with experienced professionals during the focus groups, guidelines for these problems were also developed. These guidelines can be used by product managers as well as other roles working in the product development team to create their own strategies for overcoming software product management challenges and improving software engineering practices. Finally, the solutions and guidelines presented, when combined with a list of problems identified in previous research, will form a Software Product Management Guide– a framework currently under construction, as part of wider research, for product development teams to improve software engineering practices.	☑	☐	☐	☐	☐	☐	☑	☐
104	SOHCO: A Strategy for Constructing Efficient Teams	Software engineering is concerned with organizational issues, project management, and human behavior. In the process of constructing a work team, leadership faces the task of evaluating the talents and abilities of each professional and combining them into a cohesive unit that matches the profile of the project. This article describes the SOHCO technique, a strategy for forming work teams that calculates a score for each candidate based on the project profile, as well as a ranking for the work team whose desired objective most closely aligns with the project profile. This article presents exploratory research, an experiment, two real-world case studies, and a comparison of methods using WS and RW coefficients to assess ranking consistency and similarity. Results reveal that SOHCO makes constructing a work team more objective, reduces leadership effort, enhances the ability to evaluate new team arrangements, increases the probability of project success, and reduces training costs. As a limitation, the SOHCO method does not take sub-criteria weights into account.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
105	Simulation of Software Development Team Productivity Incorporating Social and Human Factors: A System Dynamics Model	Managing software development work teams requires planning resources and activities to complete projects and deliver products satisfactorily and successfully. Estimating project time is part of the planning stage and is mainly conducted using methods based on technical factors. However, since software development is a process involving people with high levels of interaction, it is necessary to consider non-technical factors in project management. This paper presents a simulation model to support informed decision-making during planning, considering that non-technical factors, specifically social and human factors (SHF), can affect product delivery time. From a systems perspective, software development is a complex system. Therefore, System Dynamics (SD) modeling based on the rework cycle archetype is used. The resulting model allows for analyzing the productivity of software development teams, integrating three key SHF: communication, leadership, and teamwork. The generated burndown charts are used to demonstrate that the model constitutes a basal structure oriented to understand the productivity behavior of work teams. By taking a systemic approach, the model introduces new ways to identify dynamic behaviors and facilitates the prediction of possible scenarios in the evolution of tasks, which helps work teams manage their risks. Additionally, leadership strategies in accordance with the team's status and a good perception of communication can reduce rework and improve the ability to deliver software products on time. To the best of our knowledge, the literature reported on approaches that holistically integrate these elements is limited, which makes this proposal a significant contribution to the discipline.	☑	☐	☐	☐	☐	☐	☑	☐
106	Sentiment Analysis Based on Improved Transformer Model and Conditional Random Fields	With the rapid development of the Internet, people independently write comments with emotional characteristics on e-commerce platforms, which express consumers' emotional tendencies towards products or services from multiple perspectives. The sentiment analysis technology of product reviews has attracted more and more attention. In recent years, the Transformer model has performed well in the field of text sentiment analysis. However, in the process of text emotion classification, the Transformer basic model cannot be well obtained when the distance between words is relatively long, and there are problems such as a low accuracy rate and recall rate and too much time spent on the overall training of model construction. Using conditional random field (CRF) as a classifier can effectively solve the problem of too long text segmentation distance, reduce model training time, and increase the accuracy of text sentiment analysis. Therefore, this paper proposes a new method for text sentiment analysis combining the improved Transformer model and a conditional random field. First of all, this paper enhances the Transformer model by adapting the decoder to better suit the task of sentiment classification, and introduces an enhanced version of the Transformer model. It is then classified by combining long-short-term memory (LSTM) and CRF. The experimental results show that in the IMDB data set, the Transformer CRF model has higher accuracy, recall rate, and F1 value, reaching 85.51%, 83.77%, and 85.06%, respectively. Compared with other methods, the results of evaluation indicators further verify that the text method has better recognition performance and generalization ability, and at the same time, it has a better understanding of customers' emotions towards a specific aspect, provides users with accurate services, effectively improves user satisfaction, and has a high use value for enterprises' business decisions.	☑	☐	☐	☐	☐	☐	☑	☐
107	Search-Based Software Test Data Generation for Path Coverage Based on a Feedback-Directed Mechanism	Automatically generating test cases by evolutionary algorithms to satisfy the path coverage criterion has attracted much research attention in software testing. In the context of generating test cases to cover many target paths, the efficiency of existing methods needs to be further improved when infeasible or difficult paths exist in the program under test. This is because a significant amount of the search budget (i.e., time allocated for the search to run) is consumed when computing fitness evaluations of individuals on infeasible or difficult paths. In this work, we present a feedback-directed mechanism that temporarily removes groups of paths from the target paths when no improvement is observed for these paths in subsequent generations. To fulfill this task, our strategy first organizes paths into groups. Then, in each generation, the objective scores of each individual for all paths in each group are summed up. For each group, the lowest value of the summed up objective scores among all individuals is assigned as the best aggregated score for a group. A group is removed when no improvement is observed in its best aggregated score over the last two generations. The experimental results show that the proposed approach can significantly improve path coverage rates for programs under test with infeasible or difficult paths in case of a limited search budget. In particular, the feedback-directed mechanism reduces wasting the search budget on infeasible paths or on difficult target paths that require many fitness evaluations before getting an improvement.	☑	☐	☐	☐	☐	☐	☑	☐
108	SaaS Application Maturity Assessment Model	Software-as-a-service (SaaS), as a software delivery model, has received substantial attention from software providers and users alike. In recent years, it has become one of the most promising service delivery models in cloud computing. Many existing companies are transferring their business into the SaaS delivery model. Network vendors also migrate to a SaaS business model by offering on-demand remote IT support. This increasingly competitive landscape and the variety in markets have imposed many challenges for SaaS developers and vendors and made it difficult to find a consensus on the factors contributing to the positive performance of SaaS businesses. This paper thoroughly explains the critical success factors in the SaaS application development process. The proposed SaaS maturity model evaluates the organizations' current SaaS development methodology. The model's framework includes an assessment questionnaire, performance scale, and rating method adapted from the BOOTSTRAP algorithm. The assessment questionnaire collects information about the organization's current process, practices, and policies and calculates the organization's maturity level based on the responses. This study considers four dimensions to access the maturity level, i.e. design, architecture, business performance, and overall SaaS organization. Consequently, this work formulates a comprehensive and integrated strategy for SaaS application development maturity evaluation.	☑	☐	☐	☐	☐	☐	☑	☐
109	Project-Sized Scaffolding for Software Engineering Courses	Students can often graduate with a degree in computer science without working with legacy code bases, yet when they join the workforce they will almost certainly work on an existing project with thousands, if not millions of lines of existing code. In order to give students a realistic experience without overwhelming them, we added scaffolding to an existing open source project and used it in our third year software engineering course. We asked students to complete a series of 5 tasks, from bug fixing to feature addition, with this scaffolded project. Our scaffolding consisted of enhanced documentation, demonstration videos, compilation videos, enhanced task descriptions, and hints for task completion. After running this course project we collected feedback via a survey (n=87) and a small focus group (n=7). We found that students appreciated the realistic experience, but that they recommend further scaffolding, especially within source code, to better balance between difficulty and learning.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
110	Privacy Engineering in the Wild: Understanding the Practitioners' Mindset, Organizational Aspects, and Current Practices	Privacy engineering, as an emerging field of research and practice, comprises the technical capabilities and management processes needed to implement, deploy, and operate privacy features and controls in working systems. For that, software practitioners and other stakeholders in software companies need to work cooperatively toward building privacy-preserving businesses and engineering solutions. Significant research has been done to understand the software practitioners' perceptions of information privacy, but more emphasis should be given to the uptake of concrete privacy engineering components. This research delves into the software practitioners' perspectives and mindset, organizational aspects, and current practices on privacy and its engineering processes. A total of 30 practitioners from nine countries and backgrounds were interviewed, sharing their experiences and voicing their opinions on a broad range of privacy topics. The thematic analysis methodology was adopted to code the interview data qualitatively and construct a rich and nuanced thematic framework. As a result, we identified three critical interconnected themes that compose our thematic framework for privacy engineering "in the wild": (1) personal privacy mindset and stance, categorised into practitioners' privacy knowledge, attitudes and behaviours; (2) organizational privacy aspects, such as decision-power and positive and negative examples of privacy climate; and, (3) privacy engineering practices, such as procedures and controls concretely used in the industry. Among the main findings, this study provides many insights about the state-of-the-practice of privacy engineering, pointing to a positive influence of privacy laws (e.g., EU General Data Protection Regulation) on practitioners' behaviours and organizations' cultures. Aspects such as organizational privacy culture and climate were also confirmed to have a powerful influence on the practitioners' privacy behaviours. A conducive environment for privacy engineering needs to be created, aligning the privacy values of practitioners and their organizations, with particular attention to the leaders and top management's commitment to privacy. Organizations can also facilitate education and awareness training for software practitioners on existing privacy engineering theories, methods and tools that have already been proven effective.	☑	☐	☐	☐	☐	☐	☑	☐
111	Practicing Equity Diversity Inclusion (EDI) in Software Development Teams: A Systematic Literature Survey	Human factors in successful software projects have always been a critical element in software engineering, however, it has always been overshadowed by focusing more on technology and underlying processes. This work is inspired by the recent increasing interest from the software engineering research community in human factors and software development by leveraging and understanding some examples of human factors such as Equity, Diversity, and Inclusion (EDI) which were not given due research consideration earlier. We performed a systematic literature review (SLR) to review the state-of-the-art literature on practicing EDI in software development teams despite of country or culture. We found that evidence of comprehensive research about practicing EDI in software development teams is limited, the up-to-date majority focus is on the topic of diversity, whereas research on topics of practicing equity and inclusion in software development teams is sporadic. It is expected that investigating the impact of human factors in the context of EDI's triangle will generate new knowledge. This will allow software practitioners to understand the benefits of practicing EDI in managing software development teams as well as provide opportunities to incorporate them into the core development process activities. In the end, future research directions for EDI practices in software development teams are also identified.	☑	☐	☐	☑	☐	☐	☐	☐
112	Poster: Kotlin Assimilating the Android Ecosystem - An Appraisal of Diffusion and Impact on Maintainability	Kotlin is a language alternative to Java, introduced in 2011. It promises to address many of Java's limitations and lead to better application maintainability. In 2017, it became a first-class language for Android development with full tool support. We mined a dataset of 2708 Android applications on which we based our study. Our empirical assessment of the diffusion of Kotlin in Android app development shows that it is now used in around 40% of projects. Kotlin adoption has a significant positive effect on code maintainability metrics and in popularity among end-users and developers. Overall, Kotlin appears to be successfully fulfilling its promise of being a better Java for Android development.	☑	☐	☐	☐	☐	☐	☑	☐
113	PBL Student Assessment: Consistency of Different Evaluation Methods in a Computing Faculty	Contribution: This article analyzes student assessment within a computing faculty employing a full project-based learning (PBL) approach. Examining 2078 final grades across 60 classes and periods, the study reveals a significant correlation between graded self-studies, exams, and projects. This result contributes to understanding the reliability and independence of diverse evaluation methods, emphasizing their effectiveness in measuring students' learning within a PBL framework. Background: This study is motivated by the need to investigate diverse learning assessment methods within a PBL setting, aiming to establish context and underline the broad applicability of a comprehensive evaluation approach in computing education. Intended Outcomes: The study seeks to provide information about the interrelationships between different assessment methods within a PBL framework. The desired outcome is a comprehensive understanding applicable across various classes and periods, emphasizing the robustness of the assessment system. Application Design: Implementing a full PBL model in a computing faculty during a two-year period, the chosen approach integrates graded self-studies, exams, and projects to ensure a well-rounded evaluation system, capturing student proficiency across diverse computing concepts. This study involved 210 first- and second-year students, 162 men and 48 women, with a mean age of 21.7 ± 2.7 years. Findings: Correlation analysis of 2078 final grades indicates consistent reliability among graded self-studies, exams, and projects. This result underscores the effectiveness of each assessment method in measuring learning within a PBL framework, highlighting the system's robust applicability in a computing educational environment.	☐	☐	☐	☐	☐	☐	☐	☐
114	Organizational Ohm's Law in Practice: Measuring Engineering Productivity	Engineering productivity measures how effectively engineering teams produce valuable outcomes during a specific time frame. In this article, we aim to quantitatively explore the use of organizational Ohm's law to measure productivity in software engineering. According to this law, organizational productivity is directly proportional to outcome–output efficiency and organizational potential, while inversely proportional to organizational resistance. We derive measurable metrics from this law, which include engineer time allocation, engineer motivation, engineer skills, outcome–output alignment, lines of code, and usage-based business impact. We propose various practical methods to obtain these metrics and explain the considerations to determine the appropriate methods. Using these metrics, we comprehensively study and document four software product development projects. We apply the law to measure the engineering productivity of the projects. Our research supports the validity of organizational Ohm's law using lines of code output and usage-based business impact outcome. Our work demonstrates that organizational Ohm's law could provide an effective and intuitive model for understanding engineering productivity challenges.	☑	☐	☐	☐	☐	☐	☑	☐
115	Optimal Fuzzy Wavelet Neural Network Based Road Damage Detection	Floods are one of the most severe and most frequent natural calamities. It causes enormous economic damage and even leads to higher mortality rates. Studies on damage detection of roads using artificial intelligence (AI) methods gained more attention currently, especially in the development of smart cities. Therefore, this study designs an optimal Fuzzy Wavelet Neural Network based Road Damage Detection (OFWNN-RDD) technique for Flooding Management. The OFWNN-RDD technique aims to exploit the remote sensing images to classify different kinds of roads. For noise removal process, the OFWNN-RDD technique utilizes Gabor filtering (GF) technique. In addition, the OFWNN-RDD technique uses the DenseNet121 model for feature vector generation with modified barnacles mating optimization (MBMO) based hyperparameter optimizer. Finally, FWNN image classification approach is used for road damage detection. The simulation values exhibit the supremacy of the OFWNN-RDD technique over other models with improved road damage detection accuracy of 98.56%.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
116	On the Understandability of MLOps System Architectures	Machine Learning Operations (MLOps) is the practice of streamlining and optimising the machine learning (ML) workflow, from development to deployment, using DevOps (software development and IT operations) principles and ML-specific activities. Architectural descriptions of MLOps systems often consist of informal textual descriptions and informal graphical system diagrams that vary considerably in consistency, quality, detail, and content. Such descriptions only sometimes follow standards or schemata and may be hard to understand. We aimed to investigate informal textual descriptions and informal graphical MLOps system architecture representations and compare them with semi-formal MLOps system diagrams for those systems. We report on a controlled experiment with sixty-three participants investigating the understandability of MLOps system architecture descriptions based on informal and semi-formal representations. The results indicate that the understandability (quantified by task correctness) of MLOps system descriptions is significantly greater using supplementary semi-formal MLOps system diagrams, that using semi-formal MLOps system diagrams does not significantly increase task duration (and thus hinder understanding), and that task correctness is only significantly correlated with task duration when semi-formal MLOps system diagrams are provided.	☑	☐	☐	☐	☐	☐	☑	☐
117	On the Impact of Grading on Teamwork Quality in a Software Engineering Capstone Course	Every semester, we deliver a capstone course on software engineering where students undertake a real-world project in three iterations. By the end of each iteration, students are graded in several dimensions: software quality, project management, and peer assessment. The latter is the only grade assigned individually; therefore, students who are penalized by their teams (e.g., for being perceived as low contributors to the team effort) are not severely affected. This results in little incentive for improvement, which potentially jeopardizes the overall quality of the project outcome. Envisaging to promote team cohesion, we devised a new grading rule: if the peer assessment of a student is lower than a threshold, that would be their final grade in the iteration. This paper reports the results of studying the effectiveness of the proposed rule. We recorded peer assessments over six consecutive semesters: (1) the first three as the baseline measure; (2) the semester where we introduced the new grading rule; and (3) the following two semesters, as a contrast. When the rule was first introduced, peer assessments resulted low and heavily spread at the beginning, but they consistently improved toward the end of the semester. When the instructional team already trusted the rule and explicitly emphasized its application, peer assessments consistently grew along the semester but resulted in fewer outliers. The study results show that exposing peer assessments earlier on helps promote team reflection. They also made evident the positive impact of teamwork for producing quality products in a software engineering capstone course.	☑	☑	☐	☐	☐	☐	☐	☐
118	Offsite Evaluation of Localization Systems: Criteria, Systems, and Results From IPIN 2021 and 2022 Competitions	Indoor positioning is a thriving research area, which is slowly gaining market momentum. Its applications are mostly customized, ad hoc installations; ubiquitous applications analogous to Global Navigation Satellite System for outdoors are not available because of the lack of generic platforms, widely accepted standards and interoperability protocols. In this context, the indoor positioning and indoor navigation (IPIN) competition is the only long-term, technically sound initiative to monitor the state of the art of real systems by measuring their performance in a realistic environment. Most competing systems are pedestrian-oriented and based on the use of smartphones, but several competing tracks were set up, enabling comparison of an array of technologies. The two IPIN competitions described here include only off-site tracks. In contrast with on-site tracks where competitors bring their systems on-site—which were impossible to organize during 2021 and 2022—in off-site tracks competitors download prerecorded data from multiple sensors and process them using the EvaalAPI, a real-time, web-based emulation interface. As usual with IPIN competitions, tracks were compliant with the EvAAL framework, ensuring consistency of the measurement procedure and reliability of results. The main contribution of this work is to show a compilation of possible indoor positioning scenarios and different indoor positioning solutions to the same problem.	☑	☐	☐	☐	☐	☐	☑	☐
119	Model to Cope With Requirements Engineering Issues for Software Development Outsourcing	The anticipated benefits of Software Development Outsourcing (SDO) are not achieved in case of several projects because of the issues that emanate from Requirements Engineering (RE) process. This research work presents a Requirements Engineering Practices (REP) model to cope with the customarily occurring issues of the RE process for SDO. To formulate the model, five workshops have been conducted and Root Cause Analysis has been performed by considering 43 commonly occurring SDO RE process issues, and 147 RE practices to tackle the issues. To discover the root causes for commonly transpiring issues, 5-Whys technique has been employed. The relevant RE practices that can be used to deal with the root causes, have been endorsed by applying Brainstorming technique. For the 43 frequently occurring issues, 89 root causes have been discovered. Afterwards, 124 relevant RE practices have been recommended to eradicate the root causes and hence to address the corresponding issues. Thus, REP model postulates the root causes for commonly occurring issues of the SDO RE process, maps the root causes to the best relevant RE practices to address the corresponding issues. The model has been evaluated by an expert panel and evaluation results have been analysed through Inter-Rater Reliability analysis and Analysis of Means. The REP model supports the RE process for SDO by i). evading the adoption of random and inappropriate RE practices for dealing with the common issues of the process, ii) helping to attain the expected benefits of SDO.	☑	☐	☐	☐	☐	☐	☑	☐
120	MLHOps: Machine Learning Health Operations	Machine Learning Health Operations (MLHOps) is the combination of processes for reliable, efficient, usable, and ethical deployment and maintenance of machine learning models in healthcare settings. This paper provides both a survey of work in this area and guidelines for developers and clinicians to deploy and maintain their own models in clinical practice. We cover the foundational concepts of general machine learning operations and describe the initial setup of MLHOps pipelines (including data sources, preparation, engineering, and tools). We then describe long-term monitoring and updating (including data distribution shifts and model updating) and ethical considerations (including bias, fairness, interpretability, and privacy). This work therefore provides guidance across the full pipeline of MLHOps from conception to initial and ongoing deployment. We also include a checklist to ensure thorough verification of each step in the process.	☑	☐	☐	☐	☐	☐	☑	☐
121	MaizeNet: A Deep Learning Approach for Effective Recognition of Maize Plant Leaf Diseases	The presence of various maize plant leaf diseases has significantly decreased both the quality and quantity of crop production. In order to take the appropriate steps to prevent the occurrence of plant leaf diseases, it is essential to track and recognize such infections during the planting period. However, correct recognition of numerous maize diseases is difficult to achieve because the currently employed automated solutions are operationally complicated or only effective on samples with plain backgrounds. While real-world scenarios are suffering from huge sample distortions like the effect of noise, clutter in the background, and blurring of the leaf regions that increase the complexity of the recognition procedure. To alleviate the above-listed problems, a deep learning (DL) approach called the MaizeNet is proposed for the correct localization and classification of various maize crop leaf disorders. We have presented an improved Faster-RCNN approach that utilizes the ResNet-50 model with spatial-channel attention as its base network for the computation of deep keypoints which are then localized and categorized into various classes. The proposed work is tested on a standard database named Corn Disease and Severity that contains samples from three different classes of maize plant diseases which are captured under diverse conditions such as complex background, brightness, and color and size variations. The MaizeNet model attains an average accuracy score of 97.89% along with mAP value of 0.94, which is showing the effectiveness of our approach for locating and classifying the numerous types of maize leaf infections.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
122	Lean Software Startup Practices and Software Engineering Education	In the modern economy, software drives innovation and economic growth. Studies show how software increasingly influences all industry sectors. Over the past 5 decades, software engineering has also changed significantly to advance the development of various types and scales of software products. Software engineering education plays an essential role in apprising students of software technologies, processes, and practices popular in industries. Furthermore, approaches to teaching software engineering are becoming more interdisciplinary and team-centered, comparable to startup contexts. In this PhD work, I want to answer the following research questions: (1) To what extent are software engineering trends present in software engineering education research? (2) What set of common software engineering practices employed in lean software startups is transferable to the software engineering education context? (3) What is the impact of lean startup practices on software engineering students and curricula? I utilize (1) a literature review, (2) mixed-methods approaches in gathering empirical evidence, and (3) design-based research. In the first phase of the research, I pinpoint the relevance of the lean startup in software engineering education through an extensive literature review. I gather empirical evidence on lean startup practices and assess their potential transferability to software engineering education during the second research phase. I demonstrate that the lean startup is an emerging trend in software engineering education research. I demonstrate that students can acquire soft, hard, and project management skills in a more realistic context in the introduction of the growth phase of lean startup practices throughout external course activities. I expect software engineering curricula to benefit from the model and framework that I propose and validate, thus facilitating lean startup practice transfer to software engineering curricula.	☑	☑	☐	☐	☐	☐	☐	☐
123	LAURA: Let's Arrange User Requirements Accurately	One of the most important skills in software engineering is the ability to specify clear unambiguous requirements. Students in the software engineering degree program at the University of Guayaquil present difficulties in describing user requirements (without using technical words) in an unambiguous manner. Thus, the present work proposes a serious game called LAURA (which is the acrostic of "Let's Arrange User Requirements Accurately") to support software engineering students in developing their ability to specify unambiguous user requirements in natural language. LAURA has been conceived following the EMERGO methodology. To evaluate the effectiveness of the game when improving the students' specifications and to collect their perceptions about the learning experience, we performed an experiment with 64 participants, who were divided into two groups: one that used the game and one that received a traditional training session. It reports that both groups improved their ability to specify unambiguous functional user requirements after the training sessions. However, the game group showed a higher preference for the activity experience and learning experience, considering LAURA as a game that stimulates their learning, maintaining attention with non-tense interactions, and recognizing it as a tool for learning to specify unambiguous user requirements. All in all, the experiment demonstrates that the proposed LAURA serious game can be a useful learning tool that produces learning results similar to those obtained with blackboard class while improving its requirement specifications in an interactive way.	☑	☐	☐	☐	☐	☐	☐	☑
124	Intelligent Pattern Recognition Using Equilibrium Optimizer With Deep Learning Model for Android Malware Detection	Android malware recognition is the procedure of mitigating and identifying malicious software (malware) planned to target Android operating systems (OS) that are extremely utilized in smartphones and tablets. As the Android ecosystem endures to produce, therefore is the risk of malware attacks on these devices. Identifying Android malware is vital for keeping user data, privacy, and device integrity. Android malware detection utilizing deep learning (DL) signifies a cutting-edge system for the maintenance of mobile devices. DL approaches namely recurrent neural network (RNN) and convolutional neural network (CNN) are best in automatically removing intricate designs and behaviors in Android app data. By leveraging features such as application programming interface (API) call sequences, code patterns, and permissions, these approaches are efficiently differentiated between benign and malicious apps, even in the face of previous unseen attacks. This study presents an Intelligent Pattern Recognition using an Equilibrium Optimizer with Deep Learning (IPR-EODL) Approach for Android Malware Recognition. The purpose of the IPR-EODL approach is to properly identify and categorize the Android malware in such a way that security can be achieved. In the IPR-EODL technique, the data pre-processing step was applied to convert input data into a compatible setup. In addition, the IPR-EODL technique applies channel attention long short-term memory (CA-LSTM) methodology for the recognition of Android malware. To enhance the solution of the CA-LSTM algorithm, the IPR-EODL system employs the Equilibrium optimization (EO) algorithm for the hyperparameter tuning method. The experimentation evaluation of the IPR-EODL model can be verified on a benchmark Android malware database. The extensive results highlight the significant result of the IPR-EODL approach to the Android malware detection process.	☑	☐	☐	☐	☐	☐	☑	☐
125	Intelligent Fault Diagnosis of Bearings in Unsupervised Dynamic Domain Adaptation Networks Under Variable Conditions	Effective fault diagnosis is crucial for ensuring the safe and reliable operation of machinery. Despite satisfactory achievements of deep learning in fault diagnosis, acquiring large labeled datasets remains a challenge in practical industrial scenarios. In addition, machinery often operates under varying load or speed conditions, which leads to diverse distributions of collected samples. Consequently, a model trained on one distribution (source domain) encounters difficulties in adapting to another distribution (target domain). To solve the problem, this paper introduces unsupervised domain adaptation method, and also proposed unsupervised dynamic domain adaptation network (UDDAN) for fault diagnosis in bearings under variable working conditions. Specifically, the multi-scale dilated convolution module is integrated into Squeeze-and-Excitation ResNeXt (SE-ResNeXt) as a feature extractor, which not only allows the model to introduce an attention mechanism, but also enhances its multi-scale information extraction capability. The domain adaptation is achieved by minimizing the difference between the two distributions through maximum mean difference (MMD). Furthermore, the innovatively designed loss function enables the model to dynamically allocate weights between the source and target domain knowledge during the domain adaptation. Extensive experiments are conducted on two datasets, and the proposed method achieves a diagnostic accuracy of up to 99.16% in typical scenarios, while also demonstrating significant robustness in noisy environments.	☑	☐	☐	☐	☐	☐	☑	☐
126	Industrial Project-based Course on Service Oriented Design -Experience Sharing	An increasing number of academic courses adopt a project-based evaluation of the students' knowledge and skills. In such courses, students gather development and management experiences similar to those present in industry. In this paper, we present an overview on the Service Oriented Design Master course where an industrial project proposed in collaboration with an IT company is developed by students in teams. We share our experience concerning organizational issues to be addressed before the beginning of the course, during the course, and after the end of the course. Further, we provide some information about the current edition of the course and the shareable course material.	☑	☐	☐	☐	☐	☐	☑	☐
127	Improving ISO/IEC 29110 Basic Profile Process Comprehension for VSEs: A Colored Petri Net Visualization Approach Tailored for Auditor Views	The ISO/IEC 29110 is a software engineering standard widely adopted by Very Small Entities (VSEs), especially the Basic Profile, yet its implementation is complex, demanding expertise. VSEs often focus on delivering work products, while auditors assess compliance based on the detailed activities and tasks defined by the standard, creating a critical disconnect. This article proposes a novel self-assessment approach for VSEs using a formal Colored Petri Net (CPN) model focusing on the Basic Profile of the standard. This model visualizes and simulates work product-task relationships from an auditor's perspective, making complex processes more comprehensible. Our web-based system facilitates assessment via a yes/no questionnaire. Responses inject tokens into the CPN, generating a scoring report aligned with ISO/IEC 29110 activities and tasks. Users can trace non-compliant tasks and visualize task-work product relationships, specifically highlighting what an auditor seeks. By bridging this perspective gap, our CPN-based visualization empowers VSEs to better understand, comply with, self-assess, and ultimately achieve ISO/IEC 29110 certification more efficiently.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO				MOTIVO					
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
128	Improved Radial Movement Optimization With Fuzzy Neural Network Enabled Anomaly Detection for IoT Assisted Smart Cities	Recently, an extensive implementation of the recent Internet of Things (IoT) model has resulted in the development of smart cities. The network traffic of smart cities using IoT systems has developed rapidly and established novel cybersecurity problems later these IoT devices are linked to sensors that are directly linked to huge cloud servers. Unfortunately, IoT systems and networks can be identified as extremely exposed to security attacks that aim at service accessibility and data integrity. Additionally, the heterogeneity of data gathered in distinct IoT devices, composed of the disturbances acquired in the IoT systems, renders the recognition of anomalous performance and threatened nodes very difficult related to typical Information Technology (IT) networks. Accordingly, there is a critical requirement for reliable and effectual anomaly detection (AD) for identifying malicious data to promise that it could not be utilized in IoT lead to decision support systems (DSS). This manuscript offers an Improved Radial Movement Optimization with Fuzzy Neural Network Enabled Anomaly Detection (IRMOFNN-AD) technique for IoT Assisted Smart Cities. The main purpose of the IRMOFNN-AD algorithm lies in the accurate and automated detection of the anomalies that exist in the IoT environment. For the feature selection process, the IRMOFNN-AD technique uses the IRMO system to elect an optimum set of features. Additionally, the IRMOFNN-AD algorithm applies the FNN model for the detection and classification of anomalies. Besides, the sine cosine algorithm (SCA) has been employed for the parameter tuning of the FNN algorithm. The simulation value of the IRMOFNN-AD system has been tested on benchmark IDS datasets. The extensive results illustrate the better detection outcomes of the IRMOFNN-AD system interms of different measures.	☑	☐	☐	☐	☐	☐	☑	☐
129	Impact of Emotional Intelligence on Leadership and Team Dynamics in Agile Software Engineering Projects	Agile software engineering emphasizes collaboration, adaptability, and team cohesion, where Leadership and interpersonal dynamics are critical for project success. Emotional Intelligence (EI), the ability to recognize, understand, and manage emotions, has been theorized to enhance communication, conflict resolution, and team performance in Agile environments. However, empirical evidence on its direct and mediated impacts remains underexplored. This study examines how EI influences leadership effectiveness and team dynamics in Agile software projects, focusing on quantifying these relationships to inform actionable strategies. A quantitative approach was employed, utilizing structured surveys from 51 Agile professionals (Scrum Masters, developers, and product owners) actively engaged in frameworks such as Scrum, Kanban, and Extreme Programming (XP). Validated instruments—the Emotional Intelligence Scale (EIS), Multifactor Leadership Questionnaire (MLQ), and Team Climate Inventory (TCI) measured EI competencies, leadership behaviors, and team dynamics. Data were analyzed using correlation, regression, and mediation analyses. Results revealed significant positive correlations between EI and leadership effectiveness (r=0.504, p<0.01) and EI and team dynamics (r=0.434, p<0.01). Mediation analysis demonstrated that leadership effectiveness fully mediates the relationship between EI and team outcomes ($\beta =0.290$, p<0.01), highlighting the pivotal role of emotionally intelligent leaders in fostering trust and adaptability. Teams led by individuals with high EI reported a 20% increase in trust and marked improvements in collaboration and decision-making. These findings underscore EI as a critical catalyst for Agile success. Organizations are urged to prioritize EI development through targeted leadership training programs and integrate Digital Emotional Intelligence (DEQ) tools (Zoom and Microsoft Teams) to bridge emotional gaps in distributed teams. This study provides actionable insights for optimizing team performance, conflict resolution, and project outcomes in fast-paced software engineering environments, advocating for a human-centric approach to Agile methodologies.	☑	☐	☐	☐	☐	☐	☑	☐
130	Identifying Difficulties of Software Modeling Through Class Diagrams: A Long-Term Comparative Analysis	Software modeling is a creative activity in which software components and their relationships are identified based on customer requirements. Based on the literature, object-oriented software modeling is based on four fundamental pillars which are abstraction, encapsulation, decomposition, and inheritance. However, despite the existence of guidelines and recommendations for implementing the object-oriented approach, novice software designers do not make good design decisions, leading to inefficient designs that cannot be modifiable, understandable, or user-friendly distribute at the development level. The literature reveals that the most common difficulties faced by software designers is a lack of understanding and confusion of concepts related to the object-oriented approach, as well as difficulties in creating Unified Modeling Language diagrams, especially class diagrams. The work presented in this article uses a qualitative and quantitative approach to determine, in a group of university students, what are the most recurrent difficulties and their persistence during the time. The qualitative case study is the method that allowed to generate the documents: diagnostic and evaluation tests. Additionally, a thematic analysis was used to identify, analyze and report patterns within the data. In order to know the occurrences of the problems in the case study, as part of our quantitative approach, a comparative study was applied to compare the results obtained between the diagnostic and evaluation tests and thus establish the similarities and differences among the cases observed, through the hierarchical clustering technique. The findings of this study show us 16 difficulties identified after the qualitative analysis, while the quantitative analysis shows us the number of occurrences and their persistence over time. The difficulties reported in both analyzes focus on these three difficulties: a) Definition of attributes that could be a class, b) Classes with inadequate or insufficient behavior and, c) Incorrect use of multiplicity between classes. Each of these difficulties is analyzed in depth in this study.	☑	☐	☐	☐	☐	☐	☐	☑
131	Human-Centered Digital Sustainability: Handling Enumerated Lists in Digital Texts	In advance to the present study, the authors introduced a method which makes it possible to calculate the entropy of natural language digital texts, focusing on word-processed texts, presentations, and webpages. This entropy reveals that the more underdeveloped documents are, the more demanding their content-related modification becomes. It was also found that the time and data required to complete a modification task in an erroneous document is several times more than in its correct counterpart. This finding leads to the end-user paradox: the less trained end-users are, the more errors they make, and the modification of their documents requires more resources. To resolve these discrepancies, the present study defines the sustainability rate of natural language digital texts which calculates the losses – the waste of human resources, time, workspace, computers, energy, frustration, working in bees, losing data – generated by negligent text management. Furthermore, we present examples of how manual and enumerated lists behave to modifications in a 213-page long document and conclude from our investigations that while the waste of human and machine resources occurs repeatedly in erroneous documents, the sustainability rate remains low. To prove the necessity of correction, we cleared the sample document, which took approximately 67 hours of two experts of our research group (2×67 hours). With this method, we found that the correction of errors can be extremely demanding, but uses resources only once, and further modifications in the now correct document need only the content-required amount of time, activities, entropy, and resources, in accordance with the expectations of the person intended to update the document. To correct documents, we present the Error Recognition Model, which is proved effective and efficient in digital education. All our findings indicate that both education and industry should adapt the presented approach (1) to develop students' and end-users' computational thinking skills, (2) to manage and take advantage of errors, (3) to recognize connections between the structure of the text and the complex word processing tools, (4) to pay attention to digital sustainability – beyond hardware and software development and recycling – with a focus on the human factor. Recently, the Error Recognition Model is a reactive problem-solving approach, whose effectiveness is justified. However, the near future is to run parallel the reactive and proactive uses of this approach, while if we look far into the future, the proactive use to digital born natural language texts should dominate.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
132	FRUIT: Fuzzy Representation of Unbounded and Imprecise Terms in Non-Functional Requirements	It is challenging to verify Non-Functional Requirements (NFRs) because the traditional verification process typically uses the binary concept of true or false to determine whether a requirement has been fulfilled, leaving no space for uncertainty, even in the slightest, such as partially positive or partially negative evaluations. Therefore, it is necessary to consider fuzzy logic to provide the opportunity to verify NFRs in a more flexible manner. To do this, the present manuscript exposes a methodology called the Fuzzy Representation of Unbounded and Imprecise Terms (FRUIT). It expands the PROSE structure, which we first presented in our earlier work, to enable the defining of user NFRs in natural language (NL) when considering fuzzy logic to make more flexible the evaluation of quantitative NFRs and enable the evaluation of qualitative NFRs. To define NFRs under the FRUIT methodology, we need to select the system property, metrics (linguistic variables), and fuzzy set that enable the evaluation of the NFR. A table containing attributes, metrics, and linguistic values is supplied to aid in its definition; the stakeholders and elicitors are tasked with defining the boundaries of each linguistic value. To evaluate the FRUIT methodology, 44 group projects from the required engineering courses offered between 2022 and 2024 at the University of Guayaquil were looked at by experts. According to the results, students decreased the proportion of bad NFRs from approximately 80% to 10%. In conclusion, the results show how important the FRUIT methodology is for helping students to define NFRs in NL while making more flexible the evaluation of NFRs.	☑	☐	☐	☐	☐	☐	☑	☐
133	FedCSD: A Federated Learning Based Approach for Code-Smell Detection	Software quality is critical, as low quality, or “Code smell,” increases technical debt and maintenance costs. There is a timely need for a collaborative model that detects and manages code smells by learning from diverse and distributed data sources while respecting privacy and providing a scalable solution for continuously integrating new patterns and practices in code quality management. However, the current literature is still missing such capabilities. This paper addresses the previous challenges by proposing a Federated Learning Code Smell Detection (FedCSD) approach, specifically targeting “God Class,” to enable organizations to train distributed ML models while safeguarding data privacy collaboratively. We conduct experiments using manually validated datasets to detect and analyze code smell scenarios to validate our approach. Experiment 1, a centralized training experiment, revealed varying accuracies across datasets, with dataset two achieving the lowest accuracy (92.30%) and datasets one and three achieving the highest (98.90% and 99.5%, respectively). Experiment 2, focusing on cross-evaluation, showed a significant drop in accuracy (lowest: 63.80%) when fewer smells were present in the training dataset, reflecting technical debt. Experiment 3 involved splitting the dataset across 10 companies, resulting in a global model accuracy of 98.34%, comparable to the centralized model’s highest accuracy. The application of federated ML techniques demonstrates promising performance improvements in code-smell detection, benefiting both software developers and researchers.	☑	☐	☐	☐	☐	☐	☑	☐
134	Exploring the Integration of Generative AI Tools in Software Testing Education: A Case Study on ChatGPT and Copilot for Preparatory Testing Artifacts in Postgraduate Learning	Software testing education is important for building qualified testing professionals. To ensure that software testing graduates are ready for real-world challenges, it is necessary to integrate modern tools and technologies into the curriculum. With the emergence of Large Language Models (LLMs), their potential use in software engineering has become a focus, but their application in software testing education remains largely unexplored. This study, conducted in the Capstone Project course of a postgraduate software testing program, was carried out over two semesters with two distinct groups of students. A custom-built Travel Application limited to a web platform was used in the first semester. In the second semester, a new set of students worked with an open-source application, offering a larger-scale, multi-platform experience across web, desktop, and mobile platforms. Students initially created preparatory testing artifacts manually as a group deliverable. Following this, they were assigned an individual assignment to generate the same artifacts using LLM tools such as ChatGPT 3.5 in the first semester and Microsoft Copilot in the second. This process directly compared manually created artifacts and those generated using LLMs, leveraging AI for faster outputs. After completion, they responded to a set of assigned questions. The students’ responses were assessed using an integrated methodology, including quantitative and qualitative assessments, sentiment analysis to understand emotions, and a thematic approach to extract deeper insights. The findings revealed that while LLMs can assist and augment manual testing efforts, they cannot entirely replace the need for manual testing. By incorporating innovative technology into the curriculum, this study highlights how Generative AI can support active learning, connect theoretical concepts with practical applications, and align educational practices with industry needs.	☑	☐	☐	☐	☐	☐	☑	☐
135	Exploring the Integration of Blockchain and Distributed DevOps for Secure, Transparent, and Traceable Software Development	Distributed DevOps is a software development methodology that aims to integrate the work of development and operations teams without being bound by geographical constraints. This methodology excels in enhancing collaboration and speeding software development. However, it does suffer from a lack of security, transparency, and traceability, which can result in project delays, a lack of trust between stakeholders, and even project failure. This paper addresses these issues of Distributed DevOps by implementing Blockchain technology. In this paper, we propose a novel framework that leverages blockchain technology to address the challenges faced by Distributed DevOps. Through performance analysis, we demonstrate the effectiveness of our framework in a real-world scenario, highlighting its ability to improve transparency, traceability, and the security of the DevOps pipeline. Our findings underscore the potential of blockchain-empowered solutions in revolutionizing DevOps practices. Furthermore, this research offers a practical framework for organizations seeking to optimize their development processes by integrating blockchain technology.	☑	☐	☐	☐	☐	☐	☑	☐
136	Experimental Evaluation of Test-Driven Development With Interns Working on a Real Industrial Project	Context: There is still little evidence on differences between Test-Driven Development and Test-Last Development, especially for real-world projects, so their impact on code/test quality is an ongoing research trend. An empirical comparison is presented, with 19 participants working on an industrial project developed for an energy market software company, implementing real-world requirements for one of the company's customers. Objective: Examine the impact of TDD and TLD on quality of the code and the tests. The aim is to evaluate if there is a significant difference in external code quality and test quality between these techniques. Method: The experiment is based on a randomized within-subjects block design, with participants working for three months on the same requirements using different techniques, changed from week to week, within three different competence blocks: Intermediate, Novice and Mixed. The resulting code was verified for process conformance. The participants developed only business logic and were separated from infrastructural concerns. A separate group of code repositories was used to work without unit tests, to verify that the requirements were not too easy for the participants. Also, it was analysed if there is any difference between the code created by shared efforts of developers with different competences and the code created by participants isolated in the competence blocks. The resulting implementations had LOC order of magnitude of 10k. Results: Statistically significant advantage of TDD in terms of external code quality (1.8 fewer bugs) and test quality (5 percentage points higher) than TLD. Additionally, TDD narrows the gap in code coverage between developers from different competence blocks. At the same time, TDD proved to have a considerable entry barrier and was hard to follow strictly, especially by Novices. Still, no significant difference w.r.t. code coverage has been observed between the Intermediate and the Novice developers - as opposed to TLD, which was easier to follow. Lastly, isolating the Intermediate developers from the Novices had significant impact on the code quality. Conclusion:TDD is a recommended technique for software projects with a long horizon or when it is critical to minimize the number of bugs and achieve high code coverage.	☑	☐	☐	☐	☐	☐	☑	☐
137	Experiences with Summer Camp Communication via Discord	Teamwork and communication skills are essential for those entering the workforce, especially for software development positions. For remote development positions, the ability to work with a team and communicate remotely through a Communication tool are important skills that are generally not taught in standard university courses. In this experience report, we discuss our experience using Discord for commu-nication and collaboration during our virtual summer camp focused on teaching teamwork and game design to 27 autistic high school students. Overall, we found using Discord beneficial in many ways that we did not anticipate, including quicker instructor coordination, improved socialization, and more. Additionally, we provide recommendations for those who may want to use Discord in a similar virtual environment.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
138	Evaluating the Impact of Pair Documentation on Requirements Quality in Agile Software Development	A fundamental deliverable of the requirements engineering process is the software requirements specification (SRS) document. A reliable SRS ensures that the final product aligns with the stakeholder's expectations. Yet poorly documented requirements may lead to significant project risks, which include delays, cost overruns, and product failure. Currently, delivering a precise SRS remains a complex task which sometimes result in miscommunication and incomplete information. Such challenges are particularly seen in agile development where the iterative nature of projects requires continuous refinement of the requirements. Consequently, this empirical research aims to evaluate the impact of pair documentation on the quality and effectiveness of SRS documents within agile development teams. Twenty pairs of documentation writers worked into two groups where one group used the pair documentation, i.e., the experimental group, and the other one used conventional documentation i.e., the control group. Both groups were tasked with producing SRS documents for the same project. The outcomes are evaluated against three key criteria of completeness, ambiguity, and consistency of the documents. Our preliminary analysis indicates that the experimental group demonstrated an improvement in document quality with fewer ambiguities, more comprehensive coverage of requirements, and greater consistency across sections. Our research further suggests that collaborative efforts in pair documentation not only enhance the clarity and precision of SRS documents but also may mitigate risks associated with possible miscommunications. In addition, the findings of this study emphasize the potential of pair documentation as a strategy to deliver higher quality SRS documents in dynamic project environments.	☑	☐	☐	☐	☐	☐	☑	☐
139	Evaluating Performance and Trends in Interactive Video Retrieval: Insights From the 12th VBS Competition	This paper conducts a thorough examination of the 12th Video Browser Showdown (VBS) competition, a well-established international benchmarking campaign for interactive video search systems. The annual VBS competition has witnessed a steep rise in the popularity of multimodal embedding-based approaches in interactive video retrieval. Most of the thirteen systems participating in VBS 2023 utilized a CLIP-based cross-modal search model, allowing the specification of free-form text queries to search visual content. This shared emphasis on joint embedding models contributed to balanced performance across various teams. However, the distinguishing factors of the top-performing teams included the adept combination of multiple models and search modes, along with the capabilities of interactive interfaces to facilitate and refine the search process. Our work provides an overview of the state-of-the-art approaches employed by the participating systems and conducts a thorough analysis of their search logs, which record user interactions and results of their queries for each task. Our comprehensive examination of the VBS competition offers assessments of the effectiveness of the retrieval models, browsing efficiency, and user query patterns. Additionally, it provides valuable insights into the evolving landscape of interactive video retrieval and its future challenges.	☑	☐	☐	☐	☐	☐	☑	☐
140	Enhancing Insider Threat Detection in Imbalanced Cybersecurity Settings Using the Density-Based Local Outlier Factor Algorithm	In today's interconnected world, cybersecurity has emerged as a critical domain for ensuring the integrity, confidentiality, and availability of digital assets. Within this sphere, insider threats represent a unique and particularly insidious class of security risks, originating not from external hackers but from within the organization itself. These threats are perpetrated by individuals with inside information concerning the organization's security practices, data, and computer systems. Traditional security measures like firewalls, intrusion detection systems, and antivirus software are often inadequate for tackling insider threats effectively, owing to their focus on external threats. This inadequacy underscores the urgent need for the development and implementation of more sophisticated, targeted detection techniques for insider threats. In response to this challenge, our research introduces an innovative approach that employs the Density-Based Local Outlier Factor (DBLOF) algorithm, fine-tuned to specifically tackle the challenges posed by the imbalanced nature of the CERT r4.2 insider threat dataset. This dataset is characterized by a highly skewed distribution, with a significant majority of benign instances and only a minimal proportion of malicious activities. Conventional detection algorithms often fail to effectively identify these rare but dangerous instances, leading to a high rate of false negatives. Our methodology capitalizes on the algorithm's ability to focus on the local density deviation of data points, thereby enabling the precise identification of outliers that are indicative of potential insider threats. Through rigorous testing and validation processes, we have achieved outstanding results, with an of F-score 98%. These remarkable outcomes not only affirm the effectiveness of the DBLOF algorithm as a powerful tool for combating insider threats but also contribute valuable insights to the broader academic and professional discourse on cybersecurity. Importantly, our findings have practical implications, offering organizations actionable recommendations for boosting their internal security mechanisms against the complex and evolving landscape of insider threats.	☑	☐	☐	☐	☐	☐	☑	☐
141	Enhancing Elderly Care With Socially Assistive Robots: A Holistic Framework for Mobility, Interaction, and Well-Being	The global elderly population is projected to double by 2050, creating challenges in mobility, social isolation, and cognitive decline. Socially Assistive Robots (SARs) offer a promising solution, yet gaps remain in integrating mobility assistance with cognitive and emotional support. The research developed a comprehensive framework for SARs, emphasizing user-centered design, technological innovation, and cultural adaptability. A mixed-methods approach was employed with 200 participants aged 60+, from diverse settings, including rehab centers, retired groups, old age communities and nursing homes. Results demonstrated a 33.3% improvement in mobility scores and a 40% increase in social engagement. Health metrics revealed a 22% reduction in falls, a 15% rise in physical activity, and a 90% user satisfaction rate. Gender and age differences were evident, with males aged 71+ showing higher mobility improvements (35.6%) and females aged 71+ reporting the highest satisfaction (93%). Long-term data indicated sustained benefits, with participants retaining 80% of mobility gains and social engagement levels 38% above baseline after six months. These findings highlight the importance of culturally sensitive and user-centered SAR designs in promoting elderly well-being. The research offers a robust foundation for integrating SARs into elderly care systems, emphasizing adaptability, sustainability, and ethical considerations for future implementation.	☑	☐	☐	☐	☐	☐	☑	☐
142	Emotional Awareness in Software Development: Evaluating Software Project Managers' Perspectives	Developers' emotions significantly influence job satisfaction, motivation, productivity, and work quality throughout the software development lifecycle. Previous research highlighted the impact of human factors in various professional domains, such as decision-making, job performance, and project success. However, systematic monitoring and integration into decision-making processes within software development remain underexplored. This study investigates how software project managers approach emotional management in their projects, focusing on identifying emotional variations and their actions in response. Besides, the managers experimented with an emotional polarity-monitoring dashboard tool designed to display developers' emotional states over time. The findings reveal that emotional variations are currently identified in an ad-hoc manner, underscoring the need for a systematic approach to monitor and manage the impact of developers' emotional fluctuations on the development process. Software project managers recognized that a systematic approach such as the one provided by the tool used in this study would have the potential to boost productivity, team cohesion, decision-making, and institutional well-being, emphasizing the value of understanding developers' psychological aspects to enhance the software development process.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
143	Efficient-SwishNet Based System for Facial Emotion Recognition	Facial emotion recognition (FER) is an important research area in artificial intelligence (AI) and has many applications i.e., face authentication systems, e-learning, entertainment, deepfakes detection, etc. FER is still a challenging task due to more intra-class variations of emotions. Although prior deep learning methods have achieved good performance for FER. However, still there exists a need to develop efficient and effective FER systems robust to certain conditions i.e., variations in illumination, face angles, gender, race, background settings, and people belonging to diverse geographical regions. Moreover, a generalized model for the classification of human emotions is required to be implemented in computer systems so that they can interact with humans according to their emotions and improve their interaction. This work presents a novel light-weight Efficient-SwishNet model for emotion recognition that is robust towards the aforementioned conditions. We have introduced a low-cost, smooth unbounded above and bounded below Swish activation function in our model. Property of unboundedness helps to avoid saturation while smoothing helps in optimization and generalization of the model. Performance of the proposed model is evaluated on five diverse datasets including CK+, JAFFE, FER-2013, KDEF, and FERG datasets. We also performed a cross-corpora evaluation to show the generalizability of our model. The proposed model achieves a very high recognition rate for all datasets that prove the merit of the proposed framework for both the human facial images and stylized cartoon characters. Moreover, we conducted an ablation study with different variants of our model to prove its efficiency and effectiveness for emotions identification.	☑	☐	☐	☐	☐	☐	☑	☐
144	Effective Reskilling of Foreign-Born People at Universities - The Software Development Academy	Contribution: An intensive three-month educational program can be used for rapid integration of foreign-born people into the IT industry. A novel method for integrating industrial needs with the practical parts of a bachelor's Computer Science program. Background: The program was motivated by (1) the societal need to increase the meaningful integration of immigrants into the workforce, and (2) the demand for IT specialists in the IT labor market. Intended outcomes: An effective intensive software developer program with a high level of industrial integration and a working matching model for employment. Application design: The program consists of three different phases; recruitment of participants, training and job matching. The training is divided into six modules using five different teaching methods. An evaluation model, based on passive and active data, is implemented with fast learning loops for teachers and participants. Findings: The program has been run seven times with 263 unemployed participants of different nationalities. On average 82.6 percent of the participants found employment in the IT industry within 5 months of the course ending. Female participants are in the majority and are more successful in securing employment. The findings suggest that it was possible to rapidly prototype and deliver an advanced reskilling program within a university setting and use it as a positive method to support newcomers find meaningful work that has a direct benefit for the local IT industry, as well as for the wider society.	☑	☐	☐	☐	☐	☐	☐	☑
145	Do Automatic Test Generation Tools Generate Flaky Tests?	Non-deterministic test behavior, or flakiness, is common and dreaded among developers. Researchers have studied the issue and proposed approaches to mitigate it. However, the vast majority of previous work has only considered developer-written tests. The prevalence and nature of flaky tests produced by test generation tools remain largely unknown. We ask whether such tools also produce flaky tests and how these differ from developer-written ones. Further-more, we evaluate mechanisms that suppress flaky test generation. We sample 6 356 projects written in Java or Python. For each project, we generate tests using EvoSuite (Java) and Pynguin (Py-thon), and execute each test 200 times, looking for inconsistent outcomes. Our results show that flakiness is at least as common in generated tests as in developer-written tests. Nevertheless, existing flakiness suppression mechanisms implemented in EvoSuite are effective in alleviating this issue (71.7 % fewer flaky tests). Compared to developer-written flaky tests, the causes of generated flaky tests are distributed differently. Their non-deterministic behavior is more frequently caused by randomness, rather than by networking and concurrency. Using flakiness suppression, the remaining flaky tests differ significantly from any flakiness previously reported, where most are attributable to runtime optimizations and EvoSuite-internal resource thresholds. These insights, with the accompanying dataset, can help maintainers to improve test generation tools, give recommendations for developers using these tools, and serve as a foundation for future research in test flakiness or test generation.	☑	☐	☐	☐	☐	☐	☑	☐
146	Detecting Islamic Radicalism Arabic Tweets Using Natural Language Processing	The image of the tolerant religion of Islam has been distorted by extremists in the last two decades in many ways, such as luring teenagers into terrorist acts. Nowadays, millions of users socialize and share ideas using social media platforms such as Twitter. Typically, the ideas shared on Twitter (tweets) reach and influence many people who could simply retweet them and make them even spread faster. Unfortunately, some of these ideas are posted by extremists who share hateful Arabic content. Thus, it is very important to automate the process of controlling and monitoring hateful Arabic tweets, given that Arabic is the most widely used language in the Islamic world. In this paper, we provide a manually labeled and curated dataset of 3,000 Arabic tweets that contain hateful and non-hateful tweets. To automate the process of detecting hateful tweets, we utilize advanced Machine Learning (ML) techniques and perform sentiment analysis to capture the meaning of the Arabic words in a proper word embedding (Word2Vec). Also, we used the proposed model to classify and analyze 100,000 tweets of the last decade. The outcome of this work promotes future research on analyzing Arabic hateful speech by providing a manually labeled Arabic dataset, and the trained model (achieved 92% accuracy) which can be used as an underlying tool by governments, Internet service providers, and social media applications to detect any inflammatory tweets before they spread to a wider audience.	☑	☐	☐	☐	☐	☐	☑	☐
147	Defenses Against Evasion Attacks in the Eyes of Automotive Industry: Review From a Practical Perspective	Evasion attacks targeting perception systems, particularly image processing, pose a significant threat to the security, safety, and reliability of automated driving (AD). While a variety of defense methods have been proposed from research side, selecting a suitable one for industry use cases remains a challenge: Surveys and evaluations of state-of-the-art methods concentrate mostly on methodological instead of functional differences, and fail to draw a connection to defined industry requirements. The leaps in perception system complexity and automotive security standardization activities have widened this gap alarmingly. This survey aims to bridge concurrent research and novel practical application demands. For this we derive a concrete set of practical requirements from existing industry standards and automotive-specific requirements. This results in a novel, life-cycle inspired taxonomy and evaluation criteria for defense methods tailored to the industry applicability perspective. Lastly, we demonstrate the approach by reviewing and comparing a broad range of 78 state-of-the-art defense methods from literature in light of these requirements. We hope this work fosters research on defense method evaluation, and helps to bridge the gap between research and fast adoption in the automotive domain.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO				MOTIVO						
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
148	Dealing With Cultural Dispersion: a Novel Theoretical Framework for Software Engineering Research and Practice	Software development is fundamentally a team-driven process; researchers in software engineering have identified various human and social factors that can significantly impact it. Culture emerged as a critical element, and the diversity deriving from cultural differences can be highly impactful both positively and negatively. Despite existing knowledge about how culture influences software development, limitations persist. Most importantly, a unified and comprehensive (grounded) theory of how cultural differences influence and are managed in software development has yet to exist. This lack has two significant consequences: (1) it makes research on culture fragmented, leading to the continual definition of new concepts that do not allow state of the art to advance significantly, and (2) it reduces the ability of the research to be transferred to practitioners since there is no framework designed to be understood and used by them. To address the above-mentioned limitation, this work proposed a theoretical framework of "Dealing With Cultural Dispersion," which focuses on challenges and benefits originating from cultural differences and strategies for dealing with them. Such a framework was developed through a qualitative study using an iterative research approach, including interviews and socio-technical grounded theory for data analysis. The proposed framework was designed to reveal the tangible effects of practitioners' culture in software development, allowing software teams to (1) clearly understand the problem and (2) implement the correct strategy for addressing it. Additionally, researchers can use this framework as a foundation to (deductively) develop a more robust and comprehensive theory in this field.CCS CONCEPTS•Software and its engineering → Software organization and properties;•Social and professional topics → Cultural characteristics; Geographic characteristics.LAY ABSTRACTSoftware development is a collaborative process influenced by human and social factors. Cultural diversity has a profound impact, both positively and negatively, on this dynamic field. However, current research lacks a unified theory addressing how cultural differences affect and are managed in software development, resulting in fragmented knowledge and limited applicability to practitioners. To bridge this gap, this study introduces the "Dealing With Cultural Dispersion" framework. Developed through qualitative research methods, including interviews and socio-technical grounded theory, the framework sheds light on challenges and benefits arising from cultural differences in software teams. It serves a dual purpose: providing practitioners with a tool to understand and address cultural issues in their projects, and offering researchers a foundation to develop a more comprehensive theory in the field. By employing a holistic approach, the framework allows software teams to gain a clear understanding of cultural influences and implement tailored strategies for effective collaboration. This not only addresses current limitations in research but also empowers practitioners with a practical framework for navigating cultural diversity in the realm of software development.	☑	☐	☐	☐	☐	☐	☑	☐
149	DB-Net: A Dual-Branch Hybrid Network for Stroke Lesion Segmentation on Non-Contrast CT Images	Lesion segmentation in acute ischemic stroke (AIS) is critical for accurate diagnosis, with non-contrast CT (NCCT) serving as the primary imaging modality. However, the nature of NCCT often results in low contrast and blurred lesion boundaries, creating significant challenges for accurate segmentation. These limitations can lead to omissions, misdiagnoses, or inaccurate segmentations, directly impacting clinical assessment and timely intervention. To address these challenges, this study proposes a two-branch hybrid network combining a convolutional neural network (CNN) with a Transformer framework. The proposed architecture features the aforementioned dual-branch encoder, as well as a spatial channel difference learning module and parallel attention module. The CNN branch of the encoder performs local feature extraction whereas the Transformer branch captures global dependencies, overcoming limitations of CNN in modeling long-range relationships. The spatial channel difference learning module with multiscale parallel subnetworks enhances feature interaction between branches. Additionally, the parallel attention module in the decoder improves the detection of low-contrast lesion boundaries. Experiments on publicly available AIS datasets demonstrate the effectiveness of the proposed method, achieving a Dice score of 59.11%, with improvements of 4.2% in Dice, 10.6% in Precision, and 5.0% in Jaccard over the best-performing state-of-the-art method. The proposed approach overcomes segmentation challenges in noisy NCCT images and enhances automated methods for AIS diagnosis.	☑	☐	☐	☐	☐	☐	☑	☐
150	Curricular Hackathons for Engineering Design Learning: The Case of Engineering Design Days	Contribution: This article defines "curricular hackathons" and describes the features, challenges, and opportunities of adopting the hackathon format into engineering curricula. It is based on experience implementing a series of curricular hackathons which provide students with formative design experiences as they address ill-structured and integrative design problems. Background: Design is an essential part of engineering practice. However, undergraduate engineering students are provided few opportunities to develop the skills necessary for solving ill-structured and complex design problems prior to their final year capstone projects. Cornerstone projects and other summative activities are common attempts to introduce students to design, but often carry significant academic weight in the host course. Adapting the hackathon format into curricular elements could provide a useful pedagogy for formative design practice. Research Question: How can the hackathon format be effectively adapted into curricular activities? Methodology: The study uses a multiple case study research design to extract insight from 12 implementations of curricular hackathons in engineering at the University of Waterloo. The twelve cases were analyzed through two methods: 1) a synthesis of eight previous publications and 2) an interview study of 12 instructors involved in their design and implementation. Findings: Adapting the popular hackathon format to a curricular setting requires several adaptations to maximize the impact on students. Curricular hackathons: are a short, high intensity social experience; that guide students through the design-build-test cycle of a design problem, include opportunities for reflection; and achieve some level of integration and/or embedding in a program's curriculum.	☑	☐	☐	☐	☐	☐	☐	☑
151	Cost-Sensitive Hypergraph Learning With Structure Quality Preservation for IoT Software Defect Prediction	Generative AI is revolutionizing Software Engineering (SE), as both engineers and academics embrace this technology in their work. To better leverage this technology for software generation, it is essential to propose effective IoT software defect prediction methods. However, this task is challenging due to the unclear high-order correlation underlying the data. Moreover, in real-world IoT software defect prediction applications, different types of misclassifications generally lead to distinct losses and associated costs. However, accurately determining these specific costs is often not feasible. Under such circumstances, we propose a cost-sensitive hypergraph learning method with structure quality preservation (csHLQ) to optimize the cost information and preserve the graph quality in a principled way. Due to the representational ability on high-order relationship exploring, we employ hypergraph structure instead of graph structure to model the complex correlations among the datasets. We note that if a cost-sensitive hypergraph has a high quality, its classification results may exhibit a large margin separation. Thus, csHLQ exploits the large margin cost-sensitive hypergraph while avoiding using of a cost-sensitive hypergraph with a small margin. To measure the performance of our proposed method, we performed experiments on three distinct groups of datasets, i.e., the NASA Metrics Data Program (NASA) dataset, CK metric dataset and UCI Machine Learning Repository (UCI) dataset. Experimental results and comparisons with state-of-the-art methods demonstrate the superiority of our method.	☑	☐	☐	☐	☐	☐	☑	☐
152	Coordination Value in Agile Software Development: A Multiple Case Study of Coordination Mechanisms Managing Dependencies	Background: Agile software projects involve a high degree of coordination between project members to manage complexity and frequent change. There is a need to understand what coordinating mechanisms are valuable for project and team coordination. Coordination mechanisms, such as meetings and Slack, can foster a smooth workflow but also fragment work by interrupting the focused work of developers. Objective: This study aimed to investigate valuable coordination mechanisms and how they can be balanced against the need for uninterrupted work periods. Method: We conducted 30 interviews and observed 109 meetings in five companies using agile software development methods. We used coordination-dependency mapping to identify valuable coordination mechanisms. Results: Valuable coordination mechanisms included instant messaging tools, daily stand-up meetings, boards, open work area, Scrum of Scrums, bug crush days, BizDev meetings, and Make it Happen meetings. Conclusion: We advise companies to identify valuable coordination mechanisms using coordination-dependency mapping and then to bundle, schedule, and substitute these coordination mechanisms to reduce interruptions to development work.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
153	Constructive Code Review: Managing the Impact of Interpersonal Conflicts in Practice	Code review is an activity where developers receive feedback on their code contributions from other developers. The frequent and potentially negative feedback developers receive makes code review prone to interpersonal conflicts. There is a consensus about such behavior being anti-social and leading to negative outcomes for the code, team, project, and even the company. However, these conflicts are a naturally occurring phenomenon that can lead to reaping the benefits of code review if managed well. Interpersonal conflicts in code review are not necessarily an issue to avoid, but rather to be managed. In this study, we survey developers in two companies - Adnovum - working predominantly on closed-source projects - and in Red Hat open source projects. Based on a set of 154 respondents, we have found that 77% of developers sometimes experience inter-personal conflicts in code review, even though mostly not very frequently. These conflicts pose some degree of a problem to 64% of developers. However, developers are rather successful in deriving constructive outcomes in the face of conflicts - 24% of developers report that conflicts have more positive than negative outcomes. While they are highly successful in deriving positive outcomes for code quality and maintainability, the motivation of developers and their communication and collaboration in a team has the most potential to be harmed by conflicts. The most effective strategy to have constructive rather than destructive conflicts is managing work effort and re-assigning tasks in the team to reduce stress in review. Data and materials: https://zenodo.org/records/10477537.	☑	☐	☐	☐	☐	☐	☑	☐
154	Comparing Multiple Linear Regression, Deep Learning and Multiple Perceptron for Functional Points Estimation	This study compares the performance of Pytorch-based Deep Learning, Multiple Perceptron Neural Networks with Multiple Linear Regression in terms of software effort estimations based on function point analysis. This study investigates Adjusted Function Points, Function Point Categories, Industry Sector, and Relative Size. The ISBSG dataset (version 2020/R1) is used as the historical dataset. The effort estimation performance is compared among multiple models by evaluating a prediction level of 0.30 and standardized accuracy. According to the findings, the Multiple Perceptron Neural Network based on Adjusted Function Points combined with Industry Sector predictors yielded 53% and 61% in terms of standardized accuracy and a prediction level of 0.30, respectively. The findings of Pytorch-based Deep Learning are similar to Multiple Perceptron Neural Networks, with even better results than that, with standardized accuracy and a prediction level of 0.30, 72% and 72%, respectively. The results reveal that both the Pytorch-based Deep Learning and Multiple Perceptron model outperformed Multiple Linear Regression and baseline models using the experimental dataset. Furthermore, in the studied dataset, Adjusted Function Points may not contribute to higher accuracy than Function Point Categories.	☑	☐	☐	☐	☐	☐	☑	☐
155	Comparing a LEGO® Serious Play Activity With a Traditional Lecture in Software Engineering Education	Active learning methods are needed to promote student motivation and facilitate the development of technical and soft skills. Previous research in software engineering education shows that LEGO® Serious Play (LSP) fully aligns with these needs. However, prior works are usually based on insufficiently robust research methods that do not include a large sample, a variety of evaluation instruments, and/or rigorous comparative methods such as randomized controlled trials, which makes it difficult to obtain reliable and solid conclusions. This article presents an original LSP activity to teach software development life cycle models and core software engineering activities, tackling learning objectives different from those addressed by the LSP activities reported in prior works. The LSP activity was validated through a cluster-randomized controlled trial involving 217 computer science students. These students were divided into a control group that received a traditional lecture and an experimental group that performed the LSP activity. The research was supported by pre and post-tests that allowed the study of the knowledge attained by the students, as well as a questionnaire to gather students' perceptions. The results indicate that the students in the experimental group learned significantly more and were more motivated than their counterparts in the control group. This leads to the conclusion that LSP-based activities such as the one reported in this article are highly effective in terms of knowledge acquisition and motivation to teach some software engineering topics compared to traditional lectures.	☑	☐	☐	☐	☐	☐	☐	☑
156	Collaborative Working Spheres for Global Software Development Education During the COVID-19 Pandemic: An International Experience	This paper presents our experience implementing Collaborative Working Spheres (CWS) in the context of an international undergraduate course on Global Software Development (GSD) during the COVID-19 pandemic. Many universities around the world increased their efforts in creating educational alternatives for adequately addressing the educational challenges that this pandemic has introduced. A particular case is the training of software skilled graduates to work in globally distributed environments because learning this topic requires a lot of practical work when student motivation could have been negatively affected by the COVID-19 pandemic. However, providing highly practical GSD courses during a pandemic is a challenging task for many of these universities. It is against this backdrop that we have developed an educational tool to provide CWS, enabling undergraduate students to acquire practical experience in GSD and improve their communication and teamworking skills, even during the COVID-19 pandemic. An international empirical evaluation was conducted involving students and teachers from seven universities in different countries around the world. The obtained results showed that our approach can make a significant contribution to the development of practical projects on undergraduate GSD courses with students developing their knowledge and social skills associated with this topic. The data collected on the teachers' perceptions suggested that our approach could also be useful in introducing the GSD approach at undergraduate level when social distancing is in place.	☑	☐	☐	☐	☐	☐	☐	☑
157	Characterizing Team Cognition Within Software Engineering Teams in an Undergraduate Course	Contribution: The study characterizes aspects of cognitive and metacognitive dimensions of team cognition of software development teams in educational settings. Background: The software development industry requires software engineers and developers to work in teams; for this, there is substantial research on teamwork in the context of the organization. However, little is known about it in the context of educational settings, where there is scant research on teamwork in engineering and computing projects. Research Questions: How do students enact teamwork cognitive engagement and metacognitive regulation in the context of a systems analysis and design course? Methodology: The participants were 127 undergraduate students in a systems development course organized into 26 teams. Qualitative categories were derived from a content analysis based on recorded teamwork sessions, which were then quantized and visualized for pattern identification. Findings: Results provide each team's overall cognitive engagement coefficient and metacognitive regulation coefficient. The findings also describe three clusters, each with a description of a selected team to provide further insights into the identified patterns.	☑	☐	☐	☐	☐	☐	☐	☑
158	Characterizing Software Maintenance Meetings: Information Shared, Discussion Outcomes, and Information Captured	A type of meeting that has been understudied in the software engineering literature to date is what we term the software maintenance meeting: a regularly scheduled team meeting in which emergent issues are addressed that are usually out of scope of the daily stand-up but not necessarily challenging enough to warrant an entirely separate meeting. These meetings tend to discuss a wide variety of topics and are crucial in keeping software development projects going, but little is known about these meetings and how they proceed. In this paper, we report on a single exploratory case study in which we analyzed ten consecutive maintenance meetings from a major healthcare software provider. We analyzed what kind of information is brought into the discussions held in these meetings and how, what outcomes arose from the discussions, and what information was captured for downstream use. Our findings are varied, giving rise to both practical considerations for those conducting these kinds of meetings and new research directions toward further understanding and supporting them.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
159	Characterizing Social and Human Factors in Software Development Team Productivity: A System Dynamics Approach	Software development projects demand high levels of interaction between work team members. This way, management and decision-making must be supported by analyzing the complex dynamics generated through individual interactions to complete the projects. This complexity can be addressed using system dynamics. This modeling approach studies how the structures and relationships between variables in a system interact to generate behaviors over time. It is used to understand and analyze complex systems and make informed decisions. The first step in modeling is articulating the problem. This step defines the key variables that will be included in the model. Still, the lack of a standardized procedure to select, measure, and propose causal relationships is evident. Subjectivity is often appealed to, but this could lead to inaccurate models and biased results. The challenge intensifies when it comes to qualitative variables. This study introduces a formal methodology to characterize such variables, addressing a gap in the existing literature. The use of systematic mapping and a survey-based study is proposed. The methodology is applied to characterize three social and human factors that influence the productivity of software development teams: communication, leadership, and teamwork. The results captured primary experimental research's proven definitions, measurement mechanisms, and causal relationships. This formalized approach not only fills a significant gap in system dynamics but also lays a foundation for expanding its scope to encompass additional variables. As such, it represents a substantial methodological contribution to the field.	☑	☐	☐	☐	☐	☐	☐	☑
160	Challenges in Understanding the Relationship between Teamwork Quality and Project Success in Large-Scale Agile Projects	A number of methods for large-scale agile development have recently been suggested. Much of the advice in agile methods focuses on teamwork. Prior research has established that teamwork quality influences project success both for traditional software development teams and agile teams. Further, prior studies have also suggested that teamwork quality may play out differently in large projects compared to small. We investigated the relationship between teamwork quality and project success with a survey of 196 project participants across 34 teams in four projects, replicating a previous study on single teams. The new data do not fit the previously established theoretical model, which raises several concerns. The observed effect of teamwork quality on project success operates differently across projects. We discuss possible reasons, which include disagreements on what characterises success in large-scale agile development, "concept drift" of teamwork quality factors, the possibility that interteam factors might have more influence on project success than intrateam factors, and finally, that our study design does not capture all relevant levels and functions. We conclude with a call for more studies on the quality and frequency of interaction between teams in addition to internal team factors to further advance theory and practice within large-scale agile software development.	☑	☐	☐	☐	☐	☐	☑	☐
161	Bus Factor in Practice	Bus factor is a metric that identifies how resilient is the project to the sudden engineer turnover. It states the minimal number of engineers that have to be hit by a bus for a project to be stalled. Even though the metric is often discussed in the community, few studies consider its general relevance. Moreover, the existing tools for bus factor estimation focus solely on the data from version control systems, even though there exists other channels for knowledge generation and distribution. With a survey of 269 engineers, we find that the bus factor is perceived as an important problem in collective development, and determine the highest impact channels of knowledge generation and distribution in software development teams. We also propose a multimodal bus factor estimation algorithm that uses data on code reviews and meetings together with the VCS data. We test the algorithm on 13 projects developed at JetBrains and compared its results to the results of the state-of-the-art tool by Avelino et al. against the ground truth collected in a survey of the engineers working on these projects. Our algorithm is slightly better in terms of both predicting the bus factor as well as key developers compared to the results of Avelino et al. Finally, we use the interviews and the surveys to derive a set of best practices to address the bus factor issue and proposals for the possible bus factor assessment tool.	☑	☑	☐	☐	☐	☐	☐	☐
162	Building Bridges Instead of Putting Up Walls: An Educational Tool to Facilitate Instructors in Adopting Active Learning Methodologies for Teaching Programming	Teaching programming is a challenging task, as it requires instructors to guide students in developing complex skills such as real-world abstraction, problem-solving, and logical reasoning. However, the traditional teaching approach is often ineffective in achieving these objectives. Evidence suggests that Active Learning Methodologies (ALMs) can provide a more conducive environment for skill and competency development. Nonetheless, instructors' adoption rate of ALMs remains relatively low due to various barriers and factors, particularly in programming education. To assist instructors in facing this challenge, we present in this article CollabProg, an open collaborative repository designed to support instructors in identifying and selecting the appropriate ALMs for their teaching context and specific classroom needs. Additionally, CollabProg provides a set of practical guidelines, offering a step-by-step guide to assist instructors in adopting ALMs. We adopted the Design Science Research Methodology (DSRM) to systematically address the research problem and guide the development, evaluation, and evolution of CollabProg. Furthermore, we present two case studies to evaluate the acceptance and feasibility of using CollabProg from the perspective of instructors at different educational institutions in Brazil. The evidence demonstrates that CollabProg effectively supports instructors in adopting active learning methodologies while identifying limitations and opportunities for improvement. We also found that CollabProg helped instructors identify and choose suitable ALMs for their teaching context to meet their specific classroom needs. The guidelines provided by the repository were useful and highly practical for lesson planning in adopting ALMs. The adoption of CollabProg underscores the need for effective strategies to support instructors in teaching programming and motivating students to learn. These strategies are particularly important in collaborative learning contexts, where social interaction is key. CollabProg's versatility in supporting such contexts is important for successful instruction.	☑	☐	☐	☐	☐	☐	☐	☑
163	Bridging the Theory-Practice Gap in a Maintenance Programming Course: An Experience Report	This paper presents our experience in teaching a maintenance programming course with the aim of bridging the gap between theory and practice, a recurring issue in previous course offerings. To achieve this goal, we implemented active learning strategies within an active learning classroom setting and redesigned the project work. Our approach involves peer learning and teamwork activities to cover various aspects of legacy code maintenance. For the project work, we adopted an open-ended approach that allowed students to choose their legacy code projects, which could be open-source software or a previous software project they had worked on. Analysis of students' feedback and project reports highlighted the effectiveness of our approach in bridging the gap between theory and practice. We believe that our approach had the potential to enhance students' engagement and critical thinking abilities, as well as improve practical maintenance skills relevant to their future careers.	☑	☐	☐	☐	☐	☐	☐	☑
164	Branch Drift: A Visually Explainable Metric for Consistency Monitoring in Collaborative Software Development	Modern software engineering uses branch-based workflows for collaborative development on a shared codebase. A branch may represent a task or feature within the responsibilities of a specific developer or team. When branches are merged, unanticipated conflicts occur frequently. Their resolution is time-consuming and costly. Depending on the development process, a branch exists for a few hours or up to several months. With increasing time and many branches, keeping track of potential merge conflicts becomes difficult. Typically, developers are unaware of these conflicts until the branches are merged. Issues solvable through joint reviews, discussion, and collaboration are therefore identified and resolved late. It is evident that the effort required to resolve merge conflicts poses an organizational technical debt. Project managers and team leaders must be able to track, analyze, and interpret the merge conflict potential to call for timely actions. Therefore, a metric is required. This work transfers the drift metric for inconsistency analysis from model-driven software engineering to branch-based software development. The resulting branch-drift metric summarizes the merge conflict potential of a development project in a numeric value. Its calculation is configurable to fit any project's requirements. The branch-drift is based on an intuitive and explainable geometric representation, easing the analysis of conflict patterns and outliers. We evaluate the branch-drift by quantitatively analyzing and discussing the drift values, trends, and patterns for 21 large-scale open-source repositories over three months. The results of this work encourage and enable the continuous and configurable branch-drift calculation for arbitrary Git repositories. The developed tools and gathered data are publicly available.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
165	Blockchain-Based Software Effort Estimation: An Empirical Study	Context: The success or failure of any software development project significantly depends on the accuracy of its effort estimates. Software development effort estimation is the foundation for project bidding, budgeting, planning, and cost control. Problem: The literature shows that a lot of work has been done on software effort estimation. But still, there is a need for improvement in effort estimation by introducing new methodologies. The structured group-based and analogy-based effort estimations are the widely used estimation methods. Nevertheless, there are several shortcomings of using these methods such as lack of experts, lack of historical data, and biasness in expert opinion, which negatively affect the estimation results. Motivation: With the advancement of technologies, such limitations could be overcome. Such as leveraging the applicability of blockchain in several domains such as improvement in the software development process and network security. Method: In this article, we have proposed a Blockchain-Based Software Effort Estimation (BBSEE) methodology to improve the software effort estimation. We employ the proposed method using Web and blockchain technologies. Moreover, we also proposed evaluation criteria to assess the efficacy of the proposed method in terms of Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE) and percentage of successful predictions falling (PRED (25)). Result: We performed several case studies and analyses of expert opinions of 52 organizations to present the efficacy of the proposed method. Conclusion: We observe that the BBSEE method outperforms expert judgment and analogy-based effort estimation methodologies in terms of software effort estimation.	☑	☐	☐	☐	☐	☐	☑	☐
166	Big Data = Big Insights? Operationalising Brooks' Law in a Massive GitHub Data Set	Massive data from software repositories and collaboration tools are widely used to study social aspects in software development. One question that several recent works have addressed is how a software project's size and structure influence team productivity, a question famously considered in Brooks' law. Recent studies using massive repository data suggest that developers in larger teams tend to be less productive than smaller teams. Despite using similar methods and data, other studies argue for a positive linear or even super-linear relationship between team size and productivity, thus contesting the view of software economics that software projects are diseconomies of scale. In our work, we study challenges that can explain the disagreement between recent studies of developer productivity in massive repository data. We further provide, to the best of our knowledge, the largest, curated corpus of GitHub projects tailored to investigate the influence of team size and collaboration patterns on individual and collective productivity. Our work contributes to the ongoing discussion on the choice of productivity metrics in the operationalisation of hypotheses about determinants of successful software projects. It further highlights general pitfalls in big data analysis and shows that the use of bigger data sets does not automatically lead to more reliable insights.]	☑	☐	☐	☐	☐	☐	☑	☐
167	Ball Point Game: Playing or Learning Agile Project Management?	Agile project management has become essential in modern enterprises, particularly in the software industry, due to its flexibility and adaptability. This paper presents an empirical evaluation of the Ball Point Game (BPG), a non-digital serious game designed to teach Agile project management principles. The evaluation is based on three case studies conducted with undergraduate students at the University of São Paulo, aiming to explore the following aspects: ability to compare traditional methods and Agile methods; educational effectiveness, particularly regarding player experience and perceived learning; and skills development or improvement. The results indicate that the BPG effectively allows students to properly understand the pros and cons of Agile methodologies, fosters essential project management skills, and provides a positive and engaging learning experience. The BPG proved to be an effective tool for experiential learning by allowing students to go through all stages of the experiential learning cycle. This approach not only made learning more engaging, but also helped students develop essential skills for Agile project management. The study highlights the potential of educational games to bridge the gap between theoretical knowledge and hands-on activities, making them valuable tools in Agile management education. Our work has as novel element the integration of both game-based learning and experience-based learning approaches to teach Agile project management.	☑	☐	☐	☐	☐	☐	☐	☑
168	ArchHypo: Managing Software Architecture Uncertainty Using Hypotheses Engineering	Uncertainty is present in software architecture decisions due to a lack of knowledge about the requirements and the solutions involved. However, this uncertainty is usually not made explicit, and decisions can be made based on unproven premises or false assumptions. This paper focuses on a technique called ArchHypo that uses hypotheses engineering to manage uncertainties related to software architecture. It proposes formulating a technical plan based on each hypothesis' assessment, incorporating measures able to mitigate its impact and reduce uncertainty. To evaluate the proposed technique, this paper reports an application of the technique in a mission-critical project that faced several technical challenges. Conclusions were based on data extracted from the project documentation and a questionnaire answered by all team members. As a result, the application of ArchHypo provided a structured approach to dividing the architectural work through iterations, which facilitated architectural decision-making. However, further research is needed to fully understand its impact across different contexts. On the other hand, the team identified the learning curve and process adjustments required for ArchHypo's adoption as significant challenges that could hinder its widespread adoption. In conclusion, the evidence found in this study indicates that the technique has the potential to provide a suitable way to manage the uncertainties related to software architecture, facilitating the strategic postponement of decisions while addressing their potential impact.	☑	☐	☐	☐	☐	☐	☑	☐
169	Applications of Pruning Methods in Natural Language Processing	Deep neural networks (DNN) are in high demand because of their widespread applications in natural language processing, image processing, and a lot of other domains. However, due to their computational expense, over-parameterization, and large memory requirements, DNN applications often require the use of substantial model resources. This strict requirement of latency and limited memory availability are hurdles in the device deployment of these technologies. Therefore, a common idea could be to mitigate the DNN-based models' size without any performance degradation using different compression techniques. During the last few years, a great deal of progress has been made in the field of Natural Language Processing (NLP) using deep learning approaches. The objective of this research is to offer a thorough overview of the various pruning methods applied in the context of NLP. In this paper, we review several recent pruning-based schemes used for converting standard networks into their compact and accelerated versions. Traditionally, pruning is a technique for improving latency, reducing model size, and computational complexity which is a viable approach to deal with the above-mentioned challenges. In general, these techniques are divided into two main categories: structural and unstructured pruning methods. Structural pruning methods are further classified into filter, channel, layer, block, and movement pruning. Whereas, neuron, magnitude-based, and iterative pruning lie in the category of unstructured pruning. For each method, we discuss the related metrics and benchmarks. Then recent work on each method is discussed in detail, which provides insightful analysis of the performance, related applications, and pros and cons. Then, a comparative analysis is provided to analyze the differences among approaches. Finally, the paper concludes with possible future directions and some technical challenges.	☑	☐	☐	☐	☐	☐	☑	☐
170	Application of Big Data Technology to Assessments of Female Ovarian Reserve Dysfunction	This paper evaluated the potential application of big data technology to assessments of diminished ovarian reserve (DOR). The study enrolled 162 patients who underwent ovarian reserve function assessment for the first time in the Department of Ultrasound, Jiangsu Province Hospital of Chinese Medicine from January 2023 to December 2023. Patients were divided into normal ovarian reserve function (n = 68), early-stage DOR (n = 66), mid-stage DOR (n = 12), and late-stage DOR (n = 16) groups. Hadoop and Spark frameworks were used to build a big data platform, and the MLlib parallel machine learning library was used to implement three multivariate classification models—multilayer perceptron, one-vs-rest, and random forest classifiers—to classify and analyse the ovarian reserve function dataset and evaluate the platform's performance. In the big data platform, the random forest algorithm achieved the highest classification accuracy (89.47%), followed by the neural network (81.06%) and support vector machine (72.91%) methods. The random forest algorithm had the least time overhead for datasets smaller than 50 MB; for datasets exceeding 50 MB, the support vector machine algorithm had the least time overhead, followed by the random forest and neural network algorithms. The neural network algorithm's speedup ratio was lower than that of the other two algorithms for small datasets, but with increasing dataset size, its speedup ratio significantly exceeded those of the other two algorithms. The random forest algorithm showed substantial growth for large datasets.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
171	Application Maintenance Offshoring Using HCI Based Framework and Simple Multi Attribute Rating Technique (SMART)	Over the last two decades, the rapid expansion of the Internet has prompted a growing number of enterprises to deploy their work globally. Companies are increasingly reliant on software systems, which need ongoing modification, maintenance, and upgrades. The maintenance phase consumes approximately 80% of the total software budget. Hence, companies have been eagerly looking for offshore outsourcing of these software systems. Choosing the best sourcing model for software maintenance projects remains elusive and challenging due to a variety of technological, social, and political factors. This study aims to analyze application maintenance offshoring related factors and addresses its decision-making process. To achieve the study objectives, factors of two datasets are analyzed based on standard deviation, mean and mean error. The Critical Success Factors (CSFs) are examined thoroughly to explore their impact on decision-making process. Additionally, the study proposes a sourcing framework based on CSFs that uses the Human Computer Interaction (HCI) principles. This framework assists clients and vendors to evaluate the projects prior to offshoring decisions. To enhance decision-making process, a case study is conducted in the Information Technology (IT) industry and the Simple Multi Attribute Rating Technique (SMART) is applied. As the results show, SMART ranks the available options and helps in making effective offshoring decisions.	☑	☐	☐	☐	☐	☐	☑	☐
172	Analyzing Software Industry Trends to Improve Curriculum	In the present digital era, being skilled and updated on modern software development practices has become of crucial importance for software engineering graduates. Moreover, the freelancing industry has grown immensely in recent years, and individuals, more than ever before, are fascinated by the opportunities it offers and have greater assurance that it can be a successful and satisfying alternative to regular employment. Unlike others, in the case of software, industry is leading the education. This makes Software Engineering Education (SEE) additionally responsible for minimizing the gap between the skills of the graduating students and the skills needed by the employers out there. There is not any previous work available in this that focuses on the skills required to cope with the freelancing industry by graduate students and recommendations for improvements to Pakistan higher education curriculum that help produce graduates who are capable enough to get themselves employed in freelancing platforms. This study aims to dissect the software industry needs and trends related to the freelancing industry and to uncover suggestions for training in this dynamic field. The data was extracted through different freelancing platforms using the Scrapy framework of Python, and then LDA analysis was performed on the scraped data using Python to find the most trending topics in the SE field and better analyze the situation. Using LDA analysis, the dataset extracted at two distinct time periods is investigated to describe how the software industry changes from time to time. For validity, the updated data was scraped on runtime from freelancing websites. The results of the analysis are shown in different formats, and empirical findings are discussed with reference to two different time periods and in relation to previous studies.	☑	☐	☐	☐	☐	☐	☐	☑
173	An Ensemble Learning Algorithm for Cognitive Evaluation by an Immersive Virtual Reality Supermarket	Early screening for Mild Cognitive Impairment (MCI) is crucial in delaying cognitive deterioration and treating dementia. Conventional neuropsychological tests, commonly used for MCI detection, often lack ecological validity due to their simplistic and quiet testing environments. To address this gap, our study developed an immersive VR supermarket cognitive assessment program (IVRSCAP), simulating daily cognitive activities to enhance the ecological validity of MCI detection. This program involved elderly participants from Chengdu Second People's Hospital and various communities, comprising both MCI patients (N=301) and healthy elderly individuals (N=1027). They engaged in the VR supermarket cognitive test, generating complex datasets including User Behavior Data, Tested Cognitive Dimension Game Data, Trajectory Data, and Regional Data. To analyze this data, we introduced an adaptive ensemble learning method for imbalanced samples. Our study's primary contribution is demonstrating the superior performance of this algorithm in classifying MCI and healthy groups based on their performance in IVRSCAP. Comparative analysis confirmed its efficacy over traditional imbalanced sample processing methods and classic ensemble learning voting algorithms, significantly outperforming in metrics such as recall, F1-score, AUC, and G-mean. Our findings advocate the combined use of IVRSCAP and our algorithm as a technologically advanced, ecologically valid approach for enhancing early MCI detection strategies. This aligns with our broader aim of integrating realistic simulations with advanced computational techniques to improve diagnostic accuracy and treatment efficacy in cognitive health assessments.	☑	☐	☐	☐	☐	☐	☑	☐
174	An Empirical Study on Usage and Perceptions of LLMs in a Software Engineering Project	Large Language Models (LLMs) represent a leap in artificial intelligence, excelling in tasks using human language(s). Although the main focus of general-purpose LLMs is not code generation, they have shown promising results in the domain. However, the usefulness of LLMs in an academic software engineering project has not been fully explored yet. In this study, we explore the usefulness of LLMs for 214 students working in teams consisting of up to six members. Notably, in the academic course through which this study is conducted, students were encouraged to integrate LLMs into their development tool-chain, in contrast to most other academic courses that explicitly prohibit the use of LLMs.In this paper, we analyze the AI-generated code, prompts used for code generation, and the human intervention levels to integrate the code into the code base. We also conduct a perception study to gain insights into the perceived usefulness, influencing factors, and future outlook of LLM from a computer science student's perspective. Our findings suggest that LLMs can play a crucial role in the early stages of software development, especially in generating foundational code structures, and helping with syntax and error debugging. These insights provide us with a framework on how to effectively utilize LLMs as a tool to enhance the productivity of software engineering students, and highlight the necessity of shifting the educational focus toward preparing students for successful human-AI collaboration.CCS CONCEPTS• Software and its engineering → Software development techniques; • Applied computing → Education.	☑	☐	☐	☐	☐	☐	☐	☑
175	An Empirical Study of the Content and Quality of Sprint Retrospectives in Undergraduate Team Software Projects	The retrospective, or retro, is a fundamental component of the Agile process, widely used in both software engineering courses and industry. In a retro, teams come together at the end of a sprint to reflect on their team's performance. We conducted an empirical study to explore three research questions concerning retros in undergraduate team projects: (1) What do students reflect on? (2) What is the quality of their reflections? (3) How do teams' retros vary in terms of content and quality? Our study analyzed a corpus of 963 statements documented in the retros of 32 undergraduate software teams (n = 182 students) enrolled in four software engineering courses at two North American universities. A content analysis revealed that teams reflected most often on their work, communication, and collaboration practices. Nearly a third of teams' reflections focused on their general work practices, while nearly half focused on specific areas of the software development lifecycle-most prominently, pull requests, issues, and coding/testing/debugging. An analysis of the quality of teams' retro reflections showed that only 13% provided justification for a strategy to be stopped, continued, or started. An analysis of team-by-team results indicated significant differences in teams' retro content and quality. We compare these results to previous studies of retros in academia and industry, and consider their implications for software engineering education.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
176	Aligning the learning Experience in a Project-Based Course: lessons learned from the Redesign of a Programming Lab	In teaching and training the next generation of software engineers, programming labs with students working together in small groups provide the opportunity to obtain hands-on experience for software projects involving multiple developers. However, more than other types of courses, programming labs face some challenges in providing a similar learning outcome for all students. Based on feedback and own experience from various iterations of the programming lab at TU Dortmund University, we identified that the learning ex-perience varies significantly due to heterogeneous prior knowledge, experience levels, and personality traits of both students and tutors. In this experience report, we present our approach towards aligning the learning experience by applying three different didactic im-provements based on well-studied concepts: (1) the idea of worked-out examples is transferred to teaching the software development process by providing a small software application with all corre-sponding artefacts like diagrams, program code and documentation, focusing on their relationships and development activities. (2) Goal-oriented and structured learning is used to define learning outcomes for every group meeting as a common ground, while audience re-sponse systems are utilized to motivate the attendance and allow students to self-reflect on their knowledge and competence level. (3) We harmonize the role of tutors by holding dedicated teaching workshops for tutors' responsibilities in the programming lab. The different approaches are evaluated based on surveys for stu-dents and tutors over three iterations of the programming lab at TU Dortmund University. Both sides' positive responses and feedback resulted in an enumeration of lessons learned as recommendations and support for other similar courses.	☐	☐	☐	☐	☐	☐	☐	☐
177	Agile Approaches for Cybersecurity Systems, IoT and Intelligent Transportation	To adapt to the rapidly increasing vulnerabilities in software products and cyber threats that exploit them, security professionals are actively working with software developers to produce more secure systems. In software development, agile methods are increasingly adopted in critical software projects where security risks are prominent challenges. This adoption stems from the fact that agile methods are highly iterative and support delivering services and products in smaller batches which allows security professionals to seamlessly integrate software development security activities with agile methodologies. In addition, the iterative nature of agile software development encourages frequent inspections, tests, and patching of software systems to mitigate cybersecurity risks and vulnerabilities. Considering the massive growth of the Internet of Things (IoT) and Intelligent Transportation Systems (ITS) products, the challenge of software development while addressing the security and safety concerns of these devices will continue to increase. This paper presents a comprehensive and detailed review of agile software development in the context of IoT, ITS, and their cybersecurity and risk challenges. Furthermore, we provide a systematic comparison of the reviewed literature based on a set of defined criteria. Finally, we provide a broader outlook and an outline for designing future security-enhanced agile software development solutions for IoT and ITS systems.	☑	☐	☐	☐	☐	☐	☑	☐
178	Adopting Scrum in Hybrid Settings, in a University Course Project: Reflections and Recommendations	Agile software development (ASD) methods such as Scrum promote collocated team collaboration. However, adopting Scrum in hybrid work environments can pose challenges. Therefore, understanding the positive aspects and challenges of adopting Scrum in hybrid settings and the relevant agile tools and practices is important. Software engineering education should also prepare students with the necessary knowledge and skills for implementing ASD methods in hybrid settings. This study provides insight into implementing Scrum in a hybrid setting, in a course project, discussing its positive aspects, challenges, and relevant project management and communication tools. We used a questionnaire to collect feedback from 12 scrum teams, made up of 59 master's degree students at the University of Oulu, who developed student well-being prototypes in four sprints. We applied thematic analysis to analyze the teams' feedback. We found 16 challenges, 12 positive aspects, and varying ASD practices and tools. Key challenges included scheduling conflicts, communication issues, and estimating efforts. Discord and Microsoft Teams were popular for communication, while JIRA and Trello were favored for agile project management. This study complements software engineering literature with new empirical findings. It reveals practices and tools applicable to software practitioners and start-up teams for effectively adopting scrum in hybrid settings. It also contributes by illustrating the adoption of scrum course projects in hybrid settings and providing recommendations to enhance the design of similar courses. It identifies research gaps, such as the need for tools and practices to improve collaboration and the potential of asynchronous chats in hybrid environments.	☑	☐	☐	☐	☐	☐	☑	☐
179	Additional Detection of Clones Using Locally Sensitive Hashing	Today, there are many methods for detecting blocks with repetitions and redundancy in the program code. But mostly they turn out to be dependent on the programming language in which the software is developed and try to detect complex types of repeating blocks. Therefore, the goal of the research was to develop a language-independent repetition detector and expand its capabilities. In the development and operation of the language-independent incremental repeater detector, it was decided to conduct experiments for five open source systems for evaluation using the industrial detector SIG (Software Improvement Group), including the use of a tool syntactic analysis. But there was the question of extending the algorithm for additional detection of duplication and redundancy in the code, which was proposed by Hammel, and how improvements can be made to achieve independence from the programming language. Particular attention was paid to the empirical results presented in the original study, as their effectiveness is questionable. The main parameters that were considered when creating the index for LIIRD (Language-independent incremental repeat detector) and its expansion of the LSH (locally sensitive hashing): measuring time, memory and creating an incremental step. Based on the results of experiments conducted by the authors of Hammel's work, there was a motivation to develop an extended approach. The idea of this approach is that according to the original study, the operation of calculating the entire block index with repeats and redundancy from scratch is very time consuming. Therefore, it is proposed to use LSH to obtain an effective assessment of the similarity of software project files.	☑	☐	☐	☐	☐	☐	☑	☐
180	A Unified Mechanism for Cloud Scheduling of Scientific Workflows	Scheduling plays a vital role in the efficient utilization of the available resources in clouds. This paper investigates the capabilities of the current scheduling algorithms of WorkFlowSim framework for processing scientific workflows. These investigations used four different sizes of workloads each, for, five well-known workflows. It was revealed that none of the existing algorithms is capable of efficiently executing all the four sizes of workload for the complete set of workflows. Different algorithms performed better, when they were applied to various workloads of a particular workflow. This fact was used in developing an improved unified mechanism, which is capable of using an existing algorithm that performed well in the past, against the given workload. Evaluation results showed that the proposed mechanism improved over the existing algorithms for 4 out of 5 workflows (Epigenomics, Inspiral, Cyber Shake, and Montage), when tested against an aggregated load of all sizes, in terms of simulation time. For the workflow named SIPHT, however, it responded exactly the same as Max-Min algorithm. The minimum and maximum improvements, against the existing best and worst algorithms, in percentage, for Epigenomics, Inspiral, SIPHT, Cyber Shake and Montage were 16–63, 30–68, 0–69, 30–68, and 9–71 in corresponding order. This work has an additional overhead in terms of a dedicated module to find and store algorithmic performance. It is, however, required once and, thus, the increase in execution time might be marginal. The future work intends to check the impact of compute time towards optimization parameters such as makespan, pricing and deadlines.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
181	A Three-Year Study on Peer Evaluation in a Software Engineering Project Course	Background: Peer evaluation in software engineering (SE) project courses enhances the learning experience of students. It also helps instructors monitor and assess both teams and individual students. Peer evaluations might influence the way individual students and teams work; therefore, the quality of the peer evaluations should be tracked through the project course. Contribution: In this article, we analyzed the quality and scoring behavior of students in peer evaluation in an undergraduate SE project course over three years. Research Questions: RQ1: What is the quality of peer evaluation of undergraduate students in a SE project course? RQ2: How do undergraduate students in an SE project course score each other? Methodology: The quality of peer evaluation (length, level of detail, etc.) and scoring of peers based on various aspects of peer evaluations of third-year students in a year-long SE project course were studied. Taking into account the grade students received at the end of the course (A, B, C, and F-calibers), peer evaluations were categorized, analyzed over time, and compared between students calibers. Findings: After analyzing 6854 peer evaluations from 193 students, it was found that the quality of peer evaluations across students was mostly consistent throughout the course. Also, it was observed that quantitative aspects of the peer evaluation were scored similarly across student calibers. However, the qualitative aspects of the peer evaluation were impacted by the caliber of students. These findings suggest that weaker students (i.e., C-caliber students) generally receive better quality peer evaluations than stronger students (i.e., A-caliber students). Finally, a preliminary analysis showed a positive connotation of emotions and sentiments found in the textual feedback delivered by students.	☐	☐	☐	☐	☐	☐	☐	☐
182	A Taxonomy of Inter-Team Coordination Mechanisms in Large-Scale Agile	In large-scale agile software development, many teams work together to achieve overarching project goals. The more teams, the greater the coordination requirements. Despite the growing popularity of large-scale agile, inter-team coordination is challenging to practice and research. We conducted a case study over 1.5 years in a large-scale software development firm to better understand which inter-team coordination mechanisms are used in large-scale agile and how they support inter-team coordination. Based on a thematic analysis of 31 interviews, 113 hours of observations, and supplemental material, we identified 27 inter-team coordination mechanisms. From this, we offer the following contributions. First, we propose a taxonomy of inter-team coordination with three categories: coordination meetings, such as communities of practice, inter-team stand-ups, and retrospectives; coordination roles, such as the program architects and the platform team; and coordination tools and artefacts, such as Slack and JIRA as well as inter-team task boards, product backlogs, and roadmaps. Second, the coordination mechanisms displayed combinations of four key characteristics, technical, organizational, physical, and social (TOPS), which form the basis of the TOPS framework to capture the multifaceted characteristics of coordination mechanisms. Technical relates to the software product and/or technical tools supporting software development. Organizational pertains to the structural aspects of the organization. Physical refers to tangible or spatial characteristics. Social captures interpersonal and community-based characteristics. Finally, the taxonomy and the TOPS framework provide a knowledge base and a structured approach for researchers to study as well as for software practitioners to understand and improve inter-team coordination in large-scale agile.	☑	☐	☐	☐	☐	☐	☑	☐
183	A Severity Assessment of Python Code Smells	Presence of code smells complicate the source code and can obstruct the development and functionality of the software project. As they represent improper behavior that might have an adverse effect on software maintenance, code smells are behavioral in nature. Python is widely used for various software engineering activities and tends tool to contain code smells that affect its quality. This study investigates five code smells diffused in 20 Python software comprising 10550 classes and analyses its severity index using metric distribution at the class level. Subsequently, a behavioral analysis has been conducted over the considered modification period (phases) for the code smell undergoing class change proneness. Furthermore, it helps to investigate the accurate multinomial classifier for mining the severity index. It witnesses the change in severity at the class level over the modification period by mapping its characteristics over various statistical functions and hypotheses. Our findings reveal that the Cognitive Complexity of code smell is the most severe one. The remaining four smells are centered around the moderate range, having an average severity index value. The results suggest that the J48 algorithm was the accurate multinomial classifier for classifying the severity of code smells with 92.98% accuracy in combination with the AdaBoost method. The findings of our empirical evaluation can be beneficial for the software developers to prioritize the code smells in the pre-refactoring phase and can help manage the code smells in forthcoming releases, subsequently saving ample time and resources spent in the development and maintenance of software projects.	☑	☐	☐	☐	☐	☐	☑	☐
184	A Median-Based Fuzzy Approach to Software Quality Evaluation	Software quality evaluation is a challenging task in software engineering. A new group decision-making evaluation model is presented in this work. The new model is based on the VIsekriterijumska optimizacija i KOmpromisno Resenje (VIKOR) technique, in which a group regret measurement and a group satisfaction measurement are provided to increase the number of reference criteria in the decision-making process. We choose the median to represent the center of the data. Based on this, an entropy-based weighting method is proposed and used to determine the weights of decision makers. A new normalized projection is explored to measure the closeness between two evaluation matrices in a Pythagorean fuzzy setting. Several experimental analyses demonstrate that the entropy-based weighting method developed in this study is superior to traditional weighting methods. The median-based data center provides support for stable decision outcomes. Four dynamic experiments are reported on in this paper: The first one shows that the decision results remain stable throughout the entire experimental range; the second one demonstrates that the proposed normalized projection measure outperforms traditional projection measure; the third one demonstrates that the newly developed VIKOR method outperforms the traditional VIKOR method; and the last one identifies the optimal range for the three parameters of the proposed comprehensive VIKOR model.	☑	☐	☐	☐	☐	☐	☑	☐
185	A Genetic Algorithm-Based Approach to Support Forming Multiple Scrum Project Teams	Forming effective teams is an essential but challenging task, especially for organizations that carry out multiple projects simultaneously, a problem known as the Multiple Team Formation (MTF) problem. The literature presents several solutions for the MTF problem, mostly modeling it as a search problem. However, the existing solutions are not suitable for Scrum projects. We addressed this gap by developing an approach composed of two main steps. First, we designed a Structured Task Model to support creating developers' profiles given their performance on past Scrum projects. Then, given a set of target projects' technology requirements and the available developers' profiles, we developed a Genetic Algorithm to form the teams for a set of target projects. We evaluated the proposed approach by comparing the teams formed by our approach with the ones formed by project managers from one organization. Our approach achieved 85% of precision when compared with the teams provided by the project managers who worked on the same target projects. We also recorded an acceptance rate of up to 75%. The significant value of precision achieved suggests that our approach can provide teams close to the project managers' expectations. In addition, our Structured Task Model offers a promising way to build technical profiles semi-automatically for Scrum developers. In future work, we intend to investigate how to complement the developers' profiles by using other types of attributes and knowledge sources.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
186	A Comprehensive Framework for Migrating to Zero Trust Architecture	Migrating to Zero Trust Architecture (ZTA) is a strategic approach to strengthen the enterprise's security postures. The shift to ZTA requires changes across the enterprise which can be challenging to achieve. Utilizing an effective framework for migrating from the old security architecture to ZTA can help ensure smooth transitioning to the Zero Trust journey. In previous research, the effective frameworks and processes for migrating to ZTA have not been achieved in an integrated and comprehensive manner. This study introduces a comprehensive framework for migrating to ZTA. The methodology of this study is to analyze and synthesize published studies relating to ZTA migration. The findings from existing knowledge construct a process-driven for the ZTA migration. In addition, the study addresses migration challenges and gaps in existing ZTA migration approaches. As a result, a comprehensive Zero Trust migration framework is developed to construct streamlined processes. The proposed framework can be used as a reference model for effective transitioning to ZTA.	☑	☐	☐	☐	☐	☐	☑	☐
187	A Biohybrid Interaction Framework for the Integration of Robots in Animal Societies	The collective behavior of animals has been traditionally studied through observation, quantitative models of behavior, and devices of low intelligence. Nowadays, the advancements in the field of robotics allow for closed-loop experiments that occur in real-time, and for artificial agents that intelligently and autonomously blend into hybrid groups of animals and robots (i.e., biohybrid groups). Such systems provide scientists with the opportunity to study the social interactions that govern collective behavior from within the group of animals, either by mimicking them or by introducing precise stimuli to elicit and study their response. In this work, we introduce an open source Biohybrid Observation and Interaction (BOBI) platform for small animals, which consists of 3 main components: 1) an experimental setup; 2) a wheeled mobile robot called LureBot; 3) their software. We demonstrate the software and hardware design aspects that make BOBI stand out compared to our previous system and similarly purposed systems. Finally, we demonstrate our methodologies and the new robot's agility, by conducting a set of preliminary experiments with rummy-nose tetra (Hemigrammus rhodostomus) fish, where we study: 1) the response of one fish swimming with a biomimetic and a non-biomimetic lure; 2) the robot behavior when the robot interacts biomimetically and in closed loop with the wall alone, with 1 fish, and with 4 fish. The experiments show that fish strongly prefer biomimetic lures, and that the robot is consistently successful in engaging in social interactions with the fish.	☑	☐	☐	☐	☐	☐	☑	☐
SOL										
188	Practicing the Extension in Software Engineering Education: an Experience Report	University Extension is an educational, cultural, and scientific process that establishes the exchange between academic and popular knowledge. Project-Based Learning (PjBL) consists of allowing students to confront the issues and problems of the world through a real project. PjBL has been adopted in Software Engineering (SE) teaching to motivate and engage students in studying and applying processes and methods. In this work, the university extension was integrated with the PjBL in teaching SE in a postgraduate course to bring students closer to real practices of software development, meeting a demand from the State of notorious value for society. Interviews were conducted with stakeholders, artifacts were delivered, and students received feedback from the professor and the stakeholders. Through a questionnaire, the students provided feedback about the practical activity, the learning method, and their perceptions regarding their evolution and dedication to group work. We can highlight the following findings: (i) the use of PjBL together with extension and the development of a real project was promising for the evolution of learning; (ii) greater motivation of the students, who evaluated: (iii) the learning method as satisfactory, and (iv) the practical activity and their evolution of learning as good and excellent. Our main contribution is the lessons learned that might guide future experiences that adopt PjBL in SE teaching and extension.	☑	☐	☐	☐	☑	☐	☐	☐
WoS										
189	Work-from-home impacts on software project: A global study on software development practices and stakeholder perceptions	ContextThe COVID-19 pandemic has had a disruptive impact on how people work and collaborate across all global economic sectors, including software business. While remote working is not new for software engineers, forced WFH situations come with both limitations and opportunities. As the 'new normal' for working might be based on the current state of Work-from-home (WFH), it is useful to understand what has happened and learn from that.ObjectiveThis study aims to gain insights into how their WFH arrangement impacts project management and software engineering. We are also interested in exploring these impacts in different contexts, such as startups and established companies.MethodWe conducted a global-scale, cross-sectional survey during the spring and summer 2021. Our results are based on quantitative and qualitative analysis of 297 valid responses.ResultsWe characterize the profile of WFH in both spatial and temporal aspects, together with a set of common collaborative tools and coordination and control mechanisms. We revealed some areas of project management that are relatively more challenging during WFH situations, such as coordination, communication and project planning. We also revealed a mixed picture of the perceived impact of WFH on different software engineering activities.ConclusionWFH is a situational phenomenon which can have both negative and positive impact on software teams. For practitioners, we suggest a unified approach to consider the context of WFH, collaborative tools, associated coordination and control approaches and a process that resolve those aspects that are sensitive to physical interaction.	☑	☐	☐	☐	☐	☐	☑	☐
191	What Leads to a Confirmatory or Disconfirmatory Behavior of Software Testers?	Background: The existing literature in software engineering reports adverse effects of confirmation bias on software testing. Confirmation bias among software testers leads to confirmatory behavior, which is designing or executing relatively more specification consistent test cases (confirmatory behavior) than specification inconsistent test cases (disconfirmatory behavior). Objective: We aim to explore the antecedents to confirmatory and disconfirmatory behavior of software testers. Furthermore, we aim to understand why and how those antecedents lead to (dis)confirmatory behavior. Method: We follow grounded theory method for the analyses of the data collected through semi-structured interviews with twelve software testers. Results: We identified twenty antecedents to (dis)confirmatory behavior, and classified them in nine categories. Experience and Time are the two major categories. Experience is a disconfirmatory category, which also determines which behavior (confirmatory or disconfirmatory) occurs first among software testers, as an effect of other antecedents. Time Pressure is a confirmatory antecedent of the Time category. It also contributes to the confirmatory effects of antecedents of other categories. Conclusion: The disconfirmatory antecedents, especially that belong to the testing process, e.g., test suite reviews by project team members, may help circumvent the deleterious effects of confirmation bias in software testing. If a team's resources permit, the designing and execution of a test suite could be divided among the test team members, as different perspectives of testers may help to detect more errors. The results of our study are based on a single context where dedicated testing teams focus on higher levels of testing. The study's scope does not account for the testing performed by developers. Future work includes exploring other contexts to extend our results.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
192	Vocabulary Acquisition through Thesaurus Modelling within ESP Course for Engineering Students	The paper is intended to reveal positive experience of applying a newly -elaborated thesaurus modelling technology which represents an effective tool of engineering pedagogy process involving comprehensive learning of the professional vocabulary. The participants of the research were second -year engineering students of the mineral resources specializations in St. Petersburg Mining University. Conventional -based way of learning English professional terminology and vocabulary acquisition through thesaurus modelling within ESP course were used in control and experimental groups accordingly during the academic year of 2022-2023 which entails writing assignments accomplishment and final essay assessment. The gathered data were manually put into the SPSS software and analyzed statistically with the chi-square test output with identification of p -value. The purpose of the study was to determine what impact thesaurus modelling technology has on the mastering of such general competences as communication, project development and implementation, teamwork and leadership, critical thinking, self -regulation while teaching professional vocabulary to engineering students within ESP course. The results of the final essay assessment as well as the results of the general competences questionnaire proved to be statistically effective and revealed uneven distribution. The interpretation of the results showed that although communication skills and self -regulation capacity have been mastered almost equally by students of both groups, other general competences as project development and implementation, teamwork and leadership have been developed by the experimental group much more profoundly than the control group.	☑	☐	☐	☐	☐	☐	☑	☐
193	USING DESIGN-BASED RESEARCH TO LAYER CAREER-LIKE EXPERIENCES ONTO SOFTWARE DEVELOPMENT COURSES	Aim/Purpose This research aims to describe layering of career-like experiences over existing curriculum to improve perceived educational value. Background Feedback from students and regional businesses showed a clear need to increase student's exposure to career-like software development projects. The initial goal was to develop an instructor-optional project that could be used in a single mid-level programming course; however, the pilot quickly morphed into a multi-year study examining the feasibility of agile projects in a variety of settings. Methodology Over the course of four years, an agile project was honed through repeated Design Based Research (DBR) cycles of design, implementation, testing, communication, and reflective analysis. As is common with DBR, this study did not follow single methodology design; instead, analysis of data coupled with review of literature led to exploration and testing of a variety of methodologies. The review phase of each cycle included examination of best practices and methodologies as determined by analysis of oral and written comments, weekly journals, instructor feedback, and surveys. As a result of participant feedback, the original project was expanded to a second project, which was tested in another Software Engineering (SE) course. The project included review and testing of many academic and professional methodologies, such as Student Ownership of Learning, Flipped Classroom, active learning, waterfall, agile, Scrum, and Kan-ban. The study was homogenous and quasi-experimental as the population consisted solely of software engineering majors taking required courses; as based on validity of homogenous studies, class sizes were small, ranging from 8 to 20 students. Close interactions between respondents and the instructor provided interview-like settings and immersive data capture in a natural environment. Further, the iterative development practices of DBR cycles, along with the inclusion of participants as active and valued stakeholders, was seen to align well with software development practitioner practices broadly known as agile. Contribution This study is among the first to examine layering a career-like software development project on top of a course through alteration of traditional delivery, agile development, and without supplanting existing material. Findings In response to industry recommendations for additional career-like experiences, a standalone agile capstone-like project was designed that could be layered over an existing course. Pilot data reflected positive perceptions of the project, although students did not have enough time to develop a working prototype in addition to completing existing course materials. Participant feedback led to simultaneous development of a second, similar project. DBR examination of both projects resulted in a simplified design and the ability to develop a working prototype, if and only if the instructor was willing to make adjustments to delivery. After four years, a solution was developed that is both stable and flexible. The solution met the original charge in that it required course delivery, not course material, to be adjusted. It is critical to note that when a working prototype is desired, a portion of the lecture should be flipped allowing more time for guided instruction through project-focused active learning and study group requirements. The results support agile for standalone software development projects, as long as passive delivery methods are correspondingly reduced. Recommendations for Practitioners Based on the findings, implementation of a career-like software development project can be well received as long as active learning components are also developed. Multiple cycles of DBR are recommended if future researchers wish to customize instructional delivery and develop complex software development projects. Programming instructors are recommended to explore hybrid delivery to support development of agile career-like experiences. Small class sizes allowed the researchers to maintain an interview-like setting throughout the study and future studies with larger classes are recommended to include additional subject matter experts such as graduate students as interaction with a subject matter expert was highly valued by students. Recommendations for Researchers Researchers are recommended to further examine career-like software development experiences that combine active learning with agile methods; more studies following agile and active learning are needed to address the challenges faced when complex software development is taught in academic settings. Further testing of standalone agile project development has now occurred in medium sized in person classes, online classes, independent studies, and creative works research settings; however, further research is needed. Future research should also examine the implementation of agile projects in larger class sizes. Increasing class size should be coupled with additional subject matter experts such as graduate students. Impact on Society This study addresses professional recommendations for development of agile career-like experiences at the undergraduate level. This study provides empirical evidence of programming projects that can be layered over existing curriculum, with no additional cost to the students. Initial feedback from local businesses and graduates, regarding agile projects with active learning, has been positive. The area business that refused to hire our underprepared SE graduates has now hired several. Future research Future research should explore layering agile projects over a broader range of software development courses. Feedback from hiring professionals and former students has been positive. It is also recommended that DBR be used to develop career-like experiences for online programming courses.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO				MOTIVO					
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
194	Using artificial intelligence to find design errors in the engineering drawings	Artificial intelligence is increasingly becoming important to businesses because many companies have realized the benefits of applying machine learning (ML) and deep learning (DL) in their operations. ML and DL have become attractive technologies for organizations looking to automate repetitive tasks to reduce manual work and free up resources for innovation. Unlike rule-based automation, typically used for standardized and predictable processes, machine learning, especially deep learning, can handle more complex tasks and learn over time, leading to greater accuracy and efficiency improvements. One of such promising applications is to use AI to reduce manual engineering work. This paper discusses a particular case within McDermott where the research team developed a DL model to do a quality check of complex blueprints. We describe the development and the final product of this case-AI-based software for the engineering, procurement, and construction (EPC) industry that helps to find the design mistakes buried inside very complex engineering drawings called piping and instrumentation diagrams (P&IDs). We also present a cost-benefit analysis and potential scale-up of the developed software. Our goal is to share the successful experience of AI-based product development that can substantially reduce the engineering hours and, therefore, reduce the project's overall costs. The developed solution can also be potentially applied to other EPC companies doing a similar design for complex installations with high safety standards like oil and gas or petrochemical plants because the design errors it captures are common within this industry. It also could motivate practitioners and researchers to create similar products for the various fields within engineering industry.	☑	☐	☐	☐	☐	☐	☑	☐
195	Using Active Methodologies for Teaching and Learning of Exploratory Test Design and Execution	Background: Exploratory testing has been used in the industry mainly to meet the needs of agile testing processes. In an agile scenario, this testing approach is important to facilitate changes constantly, which helps agile projects successfully. One of the ways to achieve success more easily is if the test process follows guides that provide structured activities. Problem Analysis: In this context, some studies have identified that software engineering does not understand how to apply exploratory testing approaches in a systematic way, involving both design and execution activities. After a review of the literature, it has been noticed that professionals performing exploratory testing focus only on execution activities. In an academic scenario, the literature review has evidenced that content about exploratory testing is not usually taught, and it is important that the professional acquires enough basic skills to apply it through structured activities. Purpose: In this context, this article presents an analysis and discussion of the results obtained in an experiment that applied a syllabus proposed by the author, using active methodologies to teach the Exploratory Test Design and Execution being applied and testing the approach in a systematic way. The teaching plan elaborated upon and used as part of the syllabus is based on "learning by doing" in an interactive way, that is, providing student-centered teaching in order to make such an approach more beneficial to the student and, conditioning them to obtain the expected competences in the industry. Methods: The strategies to carry out the research were as follows: (i) a review of the literature to identify gaps in the knowledge about activities performed in the exploratory testing process; then, to elaborate upon the syllabus proposed, (ii) equivalence mapping was carried out on some guides and curricula related to the research theme; (iii) a survey was conducted to identify the tools, techniques, and work products usually used by professionals in the industry; (iv) a teaching plan, based on active methodologies, was produced using the proposed syllabus; (v) then, for the application of this teaching plan in an experiment, a statistical analysis was performed using the two-tailed Student-t approach for independent samples. Results: For this, both groups (experiment and control) were analyzed using the two-tailed Student-t technique, and we noticed that significant statistical gains were reached with the use of the proposed approach, compared to the control group. All participants reported that some basic test concepts, if adopted in their courses, could contribute positively. The participation made them (students) more active in the classes provided by active pedagogical practices, especially when remote teaching was applied; this broke down barriers to communication with their classmates. Conclusion: The results show that the use of active methodologies can provide benefits to the learning of Exploratory Test Design and Execution systems as they allow the application of tools in line with real scenarios observed in software development companies; these results also encourage new studies to explore these ideas further.	☑	☐	☐	☐	☐	☐	☐	☑
196	Understanding Skill Transfer Between University and Workplace Through Reflective Practice: A Software Engineering Work-Based Learning Experience	Work-based learning (WBL) programs in Software Engineering provide students with opportunities to apply academic knowledge in the workplace. However, how students navigate the transition between knowledge and skills acquired at university and their application in the workplace remains largely unexplored. We implemented a WBL Software Engineering program in which students spend most of their time in the workplace rather than at university. Reflective student essays were used to examine how our implementation supports the integration and transfer of learning between academic and professional contexts. University modules - such as those on software design patterns and programming - offered directly applicable skills, while foundational knowledge and problem-solving frameworks supported further development. Workplace experiences reinforced understanding, provided context, and enabled practical application. However, skills such as project management, leadership, and context switching were primarily developed in the workplace, revealing gaps between academic team projects and industry needs. Difficulties in transferring some advanced technical skills also point to a disconnect between academic instruction and industry application. Our findings suggest a cyclical relationship between university and workplace learning, with each enhancing the other. We propose key considerations for WBL program design, including curriculum alignment, structured skill development, and integrated delivery. Finally, we underscore the value of reflective essays in observing and supporting skill integration.	☑	☑	☐	☐	☐	☐	☐	☐
197	Trust evaluation in virtual software development teams using BERT-based language models Evaluaci?n de confianza en equipos virtuales de desarrollo de software usando modelos de lenguajes basados en BERT	Nowadays, people from different geographical areas can be closely related thanks to advances in information and communication technologies. This has a greater impact in software development organizations where their members form virtual work teams. In these new co-located work scenarios, the construction of interpersonal trust is more complex and its impact is very relevant in the performance of software development teams. This paper presents the results of the performance evaluation of four pre-trained language models based on BERT applied to trust analysis tasks. For this work, a small dataset of 1453 comments obtained from software projects stored on Github was created. The evaluated language models achieved moderately good values, in the order of 0.84 for the F1-score metric, which augurs that with further research they could be significantly improved.	☑	☐	☐	☐	☐	☐	☑	☐
198	Towards dealing with commonly occurring requirements engineering process issues during software development outsourcing	Due to specific advantages, the volume of Software Development Outsourcing (SDO) is rapidly increasing. Because of challenges arising from the Requirements Engineering (RE) process, the anticipated benefits of SDO are not achieved in case of several projects. The objective of this research work is to recommend RE practices for addressing the commonly arising RE process issues in the case of SDO. For this reason, a thorough literature review has been undertaken, as well as two questionnaire surveys have been performed with skilled SDO industry practitioners. The surveys have been done by utilizing semi-supervised style and employing Convenience Sampling method. The 50 percent rule and a four-point Likert Scale have also been used to determine the advantages of RE practices for dealing with the issues. A comprehensive list of 147 RE practices has been extracted by conducting a Focus Group session. Furthermore, the 147 RE practices have been ranked by applying Numerical Assignment and Hundred Dollar Techniques during two Focus Group sessions. The detection and adaptation of RE practices aids in enhancing the SDO RE process, evading SDO failures, and achieving the associated SDO advantages.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
199	Toward Creating Software Architects Using Mobile Project-Based Learning Model (Mobile-PBL) for Teaching Software Architecture	Project-based learning (PBL) promotes increased levels of learning, deepens student understanding of acquired knowledge, and improves learning motivation. Students develop their ability to think and learn independently through depending on themselves in searching for knowledge, planning, exploration, and looking for solutions to practical problems. Information availability, student engagement, and motivation to learn all increase with mobile learning. The teaching process may be enhanced by combining the two styles. This paper proposes and evaluates a teaching model called Mobile Project-Based Learning (Mobile-PBL) that combines the two learning styles. The paper investigates how significantly Mobile-PBL can benefit students. The traditional lecture method used to teach the software architecture module in the classroom is not sufficient to provide students with the necessary practical experience to earn a career as software architects in the future. Therefore, the first author tested the use of the model for teaching the software architecture module at Philadelphia University's Software Engineering Department on 62 students who registered for a software architecture course over three semesters. She compared the results of using the model for teaching with those results that were obtained when using the project-based learning (PBL) approach alone. The students' opinions regarding the approach, any problems they had, and any recommendations for improvement were collected through a focus group session after finishing each semester and by distributing a survey to students to evaluate the effectiveness of the used model. Comments from the students were positive, according to the findings. The projects were well-received by the students, who agreed that it gave them a good understanding of several course ideas and concepts, as well as providing them with the required practical experience. The students also mentioned a few difficulties encountered while working on the projects, including student distraction from social media and the skills that educators and learners in higher education institutions are expected to have.	☑	☐	☐	☐	☐	☐	☐	☑
200	The Role of Team Composition in Agile Software Development Education: A Gendered Perspective	Team composition is a critical factor influencing collaboration within agile software development. This study investigates the impact of gender distribution on teamwork quality in software engineering capstone courses. We examined the experiences of 240 students organized into 40 teams during an agile project course. We analyzed two surveys, one conducted before team composition and one at the end of the project work. As much as 91% of the students chose to use the practice of conducting stand-up meetings in their project work and the majority were satisfied with the practice. Further, our analysis reveals that while women tend to engage more in design and men in programming, an increase in the proportion of women within a team correlates with a higher involvement of women in programming tasks. Our findings highlight gender differences in perceptions and experiences related to project involvement in agile software engineering courses.	☑	☐	☐	☐	☐	☐	☐	☑
201	The Optimization of Coal Pillars on Return Air Sides and the Reasonable Horizon Layout of Roadway Groups in Highly Gassy Mines	Multiple roadways are opened to solve the difficulties in gas control in the high seam working face of high gassy mines, resulting in a large width of coal pillars between roadway groups. Moreover, recovering coal pillars is challenging during stoping, which causes serious resource waste. While the size of coal pillars is optimized, the reasonable horizon layout of roadway groups is performed to ensure a fine recovery rate of coal resources, mine safety, and efficient production. Theoretical analysis was applied to calculate the stress distribution of surrounding rocks around coal pillars based on the engineering practice of the W2302 working face in the Sihe Coal Mine. The MathCAD software Prime 2.0 combined with the on-site measured stress was used to draw the stress distribution under different coal pillar widths, which can derive the stress distribution data of the floor. The vertical stress concentration factors tended to be consistent at 10 m below the coal pillars and about 12 m from the centerline of coal pillars under different coal pillar widths, this area was suitable for the layout of floor rock roadways with a stress concentration factor of about 1. Numerical simulations were used to compare the stress distributions of the roadway floor and the deformation features of surrounding rocks under different coal pillar widths. The coal pillar width was finally determined to be 45 m, which optimized the size of coal pillars. The return air roadways were arranged at 9.8 m below the coal seam combined with the strata distribution on-site. According to the industrial test on-site, the return air roadways were available during the service period, which showed reliable theoretical analysis and simulation results. A reasonable horizon layout of return air roadways can provide a scientific basis and reference for similar projects.	☑	☐	☐	☐	☐	☐	☑	☐
202	The nexus of project management approaches in sustainable development: innovative behaviors as a mechanism in the Polish financial industry	PurposeBased on the social exchange theory, the aim of the present study is to examine the effects, both direct and indirect (through sustainability-oriented innovative behaviors [SIBs]), of sustainable project leadership (SPL) on sustainable project performance (SPP). Project management approaches (PMAs) (traditional, hybrid and agile) were examined as conditional factors in the "SPL-SIBs" relationship.Design/methodology/approachThe study employs structural equation modeling based on data collected from 197 software engineering project team members working in the financial industry in Poland.FindingsThe study revealed that SPL significantly, positively affected SPP. It also provided evidence for the significant mediating impact of SIBs in the relationship between SPL and SPP and the conditional effect of agile and hybrid PMAs on the "SPL-SIBs" relationship.Originality/valueThe novelty of this work lies in introducing sustainable leadership into project management research, proposing and testing a unique and complex research framework, designing valid scales for measuring SPL and SPP, and suggesting many theoretical and empirical implications.	☑	☑	☐	☐	☐	☐	☐	☐
203	The Influence of Software Engineering Education on Developing Teamwork Skills in Both Genders	Teamwork is recognized as a highly valuable skill for any software engineer. In computer science programs it is typically developed through project-based courses, though these courses often do not adopt a gender-oriented approach. Women are underrepresented in the software industry, and there is limited knowledge about their teamwork competencies. While some studies compare teamwork skills between men and women, few have explored the impact of education on the development of this skill, particularly in relation to different aspects of teamwork. By comparing teamwork quality between genders at different program levels, we intend to gain valuable insights about students' initial teamwork skills and the effect of software engineering education over time. To do this, we have carried out an empirical study of two project-based software engineering courses in our Computer Science and Engineering program: an introductory software engineering course and a capstone course. In each case we evaluate different aspects of teamwork by applying the same peer assessment instrument along 13 semesters. Our study revealed no significant gender-based disparities within the courses under study, indicating that such differences are absent both in students' initial abilities and final competencies. What we did observe, however, is that women's teamwork assessments are more homogeneous over time than the men's, where we see significant improvements in Attitude, Motivation, Coordination and Communication between courses. Our results lead us to believe that, it takes men longer to reach the same capacity for working in teams than women, but in the end, students of both genders reach similar teamwork capabilities in all teamwork quality dimensions. Therefore, women have just as much potential to excel as software engineers as men, especially when it comes to teamwork.	☑	☑	☐	☐	☐	☐	☐	☐
204	The impact of working from home on the success of Scrum projects: A multi-method study	With the COVID-19 pandemic, Scrum teams had to switch abruptly from a traditional working setting into an enforced working from home one. This abrupt switch had an impact on software projects. Thus, it is necessary to understand how potential future disruptive events will impact Agile software teams' ability to deliver successful projects while working from home. To investigate this problem, we used a two-phased Multi-Method study. In the first phase, we uncover how working from home impacted Scrum practitioners through semi-structured interviews. Then, in the second phase, we propose a theoretical model that we test and generalize using Partial Least Squares-Structural Equation Modeling (PLS-SEM) surveying 138 software engineers who worked from home within Scrum projects. We concluded that all the latent variables identified in our model are reliable, and all the hypotheses are significant. This paper emphasizes the importance of supporting the three innate psychological needs of autonomy, competence, and relatedness in the home working environment. We conclude that the ability of working from home and the use of Scrum both contribute to project success, with Scrum acting as a mediator.(c) 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
205	THE EFFECTS OF DEVELOPER EXPERIENCE ON THE SOFTWARE ENGINEERING TEACHING-LEARNING PROCESS	Integrating theory and practice and developing professional skills required for the Software Engineering (SE) market are still challenges for universities. In this context, this article aims to investigate the effects of Developer Experience (DX) in a Project-Based Learning (PBL) approach during the SE teaching-learning process. To do so, we performed a case study on an SE subject in an Information Systems (IS) course over two semesters, using a PBL approach and a DX questionnaire. As a positive aspect, the results highlighted the development of teamwork skills and experience close to the real professional context. As a negative one, the participants reported a short time to assimilate new knowledge. Thus, it is possible to highlight two implications for teaching practice in SE: (i) checking the students' level of knowledge before starting the projects and (ii) continuous monitoring of the activities.	☑	☐	☐	☐	☐	☐	☐	☑
206	The Diversity of Approaches to Support Software Project Management in the Agile Context: Trends, Comparisons and Gaps	The literature has reported a significant number of approaches to support software development teams. Due to this diversity of approaches, the process of choosing which one is the most suitable for a given project becomes a complex task as new approaches emerge. In addition, there has been no comprehensive work on what these project management approaches are, nor what their main advantages and limitations are. Therefore, it becomes important to research the characteristics of these approaches and map them to minimize decision challenges. Therefore, the objective of this work is to identify primary studies in software engineering that present approaches to support the management of software projects in the agile context. To achieve this goal, we conducted a systematic review study and selected 65 studies on software project management approaches. From these studies, we identified eight different types of approaches, with software being the most used approaches in this context. In addition, nine project areas were identified in which these approaches focus, among which "Schedule" and "Quality" stand out as the areas with the greatest focus by the approaches. Finally, we identified that the "Planning" is the phase of the project in which the approaches place more emphasis.	☑	☐	☐	☑	☐	☐	☐	☐
207	The Design and Realization of a Self-Hosted and Open-Source Agile Internal Development Platform	Platform Engineering (PE) is a growing area of Software Engineering, with many facets, including the elusive concept of Internal Development Platforms (IDPs). They integrate various technologies and tools to support the realization of software projects, especially their managing and developing roles, fostering Agile collaboration and enhancing team productivity. In particular for the new trends of Agile remote working and for Agile applied to the Digital Transformation in the Public Sector. While proprietary IDPs abound, academic research that could naturally lead to open-source IDPs remain scarce. In particular for Scrum, a fundamental Agile methodology in the strategic area of Digital Transformation in the Public Sector, where IDP self-hosting is also appropriate. This paper addresses this gap by proposing the design and realization of the first, to the best of our knowledge, IDP supporting Scrum, being open-source and with self-hosting capabilities. The end-product of our research is the Compositional Agile System (CAS), an extendable microservice architecture-based IDP. Its design, as well as CAS, makes full use of hands-on previous successful experiences of IDP development by members of this team, specifically non-trivial tasks in Italian Public Administration and to foster remote working in University Software Engineering classes, a practice started during the pandemic but now a standard. We offer several architectural and example scenarios where CAS can be of use, concentrating on the Public Administration. Particularly relevant is the use of CAS in analyzing a pillar of Italian Digital Administration available to its citizens, i.e., Applo. The software is available as an open resource to the research community.	☑	☑	☐	☐	☐	☐	☐	☐
208	Teamwork in agile software development: A mixed-method study of gender diversity and collaboration dynamics	Context: Teamwork is an important aspect of agile software development, and the widespread adoption of agile methodologies emphasizes the need for educational approaches that prepare students for effective teamwork. Capstone courses in agile software engineering provide a valuable setting for students to experience collaborative work. However, the dynamics within these teams, particularly regarding gender, require further exploration. Objectives: This study examines teamwork experiences in an agile capstone course focusing on how team gender composition affects role distribution, teamwork quality, and gender bias. Methods: This study involved 94 teams comprising 561 students over two years (213 women and 348 men). We employed a mixed-methods approach, using four surveys (806 responses) and 12 in-depth interviews to capture ambition level, satisfaction with agile practices, and gender differences. Results: Team gender composition significantly impacted teamwork quality (TWQ), with female-dominated but gender-diverse teams showing the highest TWQ scores. Interestingly, women in token positions in male-majority teams reported unexpectedly high satisfaction levels, challenging aspects of Kanter's tokenism theory. However, women were more likely to report that team members underestimated their competence, particularly in male-majority teams. Observed gender differences included a tendency for women to engage more in design and process-oriented tasks, while men predominantly focused on programming and technical aspects such as architecture and algorithms. Conclusion: This study underscores the importance of considering team composition in relation to gender in agile teams, as we found this aspect greatly impacts team dynamics. Further, the implementation of agile practices in capstone courses yielded high overall TWQ scores, indicating significant educational value. Our findings provide insights for both software engineering education and agile practitioners.	☑	☑	☐	☐	☐	☐	☐	☐
209	Teaching Scrum with a focus on compliance assessment	The Scrum framework has gained widespread adoption in the industry for its emphasis on collaboration and continuous improvement. However, it has not reached a similar relevance in Software Engineering (SE) curricula. This work reports the experience of five editions of a SE course within an M.Sc. Degree in Computer Engineering. The course primary educational objective is to provide students with the skills to manage software development projects with Scrum. The course is based on the execution of a team project and on the definition of qualitative and quantitative means of assessment of the application of Scrum. The conduction of five editions of the course allowed us to identify several lessons learned about time budgeting and team compositions in agile student projects and its evidence of the applicability of the framework to software development courses.	☑	☑	☐	☐	☐	☐	☐	☐
210	Teaching HCI Engineering: Four Case Studies	The paper presents the work carried out at the HCI Engineering Education workshop, organised by IFIP working groups 2.7/13.4 and 13.1. It describes four case studies of projects and exercises used in Human-Computer Interaction Engineering courses. We propose a common framework for presenting the case studies and describe the four case studies in detail. We then draw conclusions on the differences between the presented case studies that highlight the diversity and multidisciplinary aspects to be taught in a Human-Computer Interaction Engineering course. As future work, we plan to create a repository of case studies as a resource for teachers.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
211	Teaching computer architecture by designing and simulating processors from their bits and bytes	Teaching computer architecture (Comp-Arch) courses in undergraduate curricula is becoming more of a challenge as most students prefer software-oriented courses. In some computer science/engineering departments, Comp-Arch courses are offered without the lab component due to resource constraints and differing pedagogical priorities. This article demonstrates how students working in teams are motivated to study the Comp-Arch course and how instructors can increase student motivation and knowledge by taking advantage of hands-on practices. The teams are asked to design and implement a 16-bit MIPS-like processor with constraints as a specific instruction set, and limited data and instruction memory. Student projects include following three phases, namely, design, desktop simulator implementation, and verification using hardware description language (HDL). In the design phase, teams develop their Comp-Arch to implement specified instructions. A range of designs resulted, e.g., (a) a processor with extensive user-defined instructions resulting in longer cycle times (b) a processor with a minimal instruction set but with a faster clock cycle time. Next, teams developed a desktop simulator in any programming language to execute instructions on the architecture. Finally, students engage in Verilog Hardware Description Language (HDL) projects to simulate and verify the data-path designed during the initial phase. Student feedback and their current understanding of the project were collected through a questionnaire featuring varying Likert scale questions, some with a ten-point scale, and others with a five- point scale. Results of the survey show that the hands-on approach increases students' motivation and knowledge in the Comp-Arch course, which is centered around computer system design principles. This approach can also be effectively extended to related courses, such as Microprocessor Design, which delves into the intricacies of creating and implementing microprocessors or central processing units (CPUs) at the hardware level. Furthermore, the present study demonstrates that interactions, specifically through peer reviews and public presentations, between students in each phase increases their knowledge and perspective on designing custom processors.	☑	☐	☐	☐	☐	☐	☐	☑
212	Swarm intelligence-based model for improving prediction performance of low-expectation teams in educational software engineering projects	Software engineering is one of the most significant areas, which extensively used in educational and industrial fields. Software engineering education plays an essential role in keeping students up to date with software technologies, products, and processes that are commonly applied in the software industry. The software development project is one of the most important parts of the software engineering course, because it covers the practical side of the course. This type of project helps strengthening students' skills to collaborate in a team spirit to work on software projects. Software project involves the composition of software product and process parts. Software product part represents software deliverables at each phase of Software Development Life Cycle (SDLC) while software process part captures team activities and behaviors during SDLC. The low-expectation teams face challenges during different stages of software project. Consequently, predicting performance of such teams is one of the most important tasks for learning process in software engineering education. The early prediction of performance for low-expectation teams would help instructors to address difficulties and challenges related to such teams at earliest possible phases of software project to avoid project failure. Several studies attempted to early predict the performance for low-expectation teams at different phases of SDLC. This study introduces swarm intelligence-based model which essentially aims to improve the prediction performance for low-expectation teams at earliest possible phases of SDLC by implementing Particle Swarm Optimization-K Nearest Neighbours (PSO-KNN), and it attempts to reduce the number of selected software product and process features to reach higher accuracy with identifying less than 40 relevant features. Experiments were conducted on the Software Engineering Team Assessment and Prediction (SETAP) project dataset. The proposed model was compared with the related studies and the state-of-the-art Machine Learning (ML) classifiers: Sequential Minimal Optimization (SMO), Simple Linear Regression (SLR), Naive Bayes (NB), Multilayer Perceptron (MLP), standard KNN, and J48. The proposed model provides superior results compared to the traditional ML classifiers and state-of-the-art studies in the investigated phases of software product and process development.	☑	☐	☐	☐	☐	☐	☐	☑
213	Student-Sponsored Projects in a Capstone Course Reflections and Lessons Learned	Undergraduate students in all disciplines sometimes struggle to see the connections between what they learn in the classroom and how it will be used in practice. Capstone courses, internships, and co-op experiences can help students bridge this gap by allowing students to put theory into practice on real-world projects and by connecting students to industry practices and partners. These experiences can better prepare students for their careers, but they may also shield students from real-world risks, such as lack of product feasibility or viability, and from the non-technical tasks of business planning and startup tasks, in order to make the academic aspects of the course manageable and scalable. In this work, we present our experience offering undergraduate Software Engineering students the opportunity to self-sponsor a year-long capstone project. Students began the course with four software product ideas, none of which had been assessed for feasibility or viability. During the academic year, students were responsible for product discovery, technology selection, and technical development following good Software Engineering practices. They also created business and marketing plans, and formed an LLC. This paper(1) describes the processes, practices and timeline we followed for a student-sponsored capstone team. We also share our lessons learned and offer suggestions for student-sponsored projects in computing capstone courses.	☐	☐	☐	☐	☐	☐	☐	☐
214	Strategies for Dealing With Software Product Management Challenges	Software product management, which involves multiple processes and responsibilities and links to many activities within the organisation, increases the success rate of IT projects. However, the adoption of product management activities and the implementation of the software product manager role itself varies between companies. This research explores the topic of software product managers' strategies for dealing with challenges in their work. The aim of the study is to identify and evaluate solutions to the main problems that affect the software product management process and to provide guidelines for dealing with them. For this purpose, a focus group method was adopted and 47 software product managers participated in 15 focus groups. This was followed by a survey to assess the comprehensibility and effectiveness of the solutions identified. The paper proposes a list of 39 solutions to the 5 most common problems in the work of product managers. Furthermore, based on the identified solutions and the results of discussions with experienced professionals during the focus groups, guidelines for these problems were also developed. These guidelines can be used by product managers as well as other roles working in the product development team to create their own strategies for overcoming software product management challenges and improving software engineering practices. Finally, the solutions and guidelines presented, when combined with a list of problems identified in previous research, will form a Software Product Management Guide- a framework currently under construction, as part of wider research, for product development teams to improve software engineering practices.	☑	☑	☐	☐	☐	☐	☐	☐
215	Story Point Estimation Using Issue Reports With Deep Attention Neural Network	Background: Estimating the effort required for software engineering tasks is incredibly tricky, but it is critical for project planning. Issue reports are frequently used in the agile community to describe tasks, and story points are used to estimate task effort. Aim: This paper proposes a machine learning regression model for estimating the number of story points needed to solve a task. The system can be trained from raw input data to predict outcomes without the need for manual feature engineering. Method: Hierarchical attention networks are used in the proposed model. It has two levels of attention mechanisms implemented at word and sentence levels. The model gradually constructs a document vector by grouping significant words into sentence vectors and then merging significant sentence vectors to create document vectors. Then, the document vectors are fed into a shallow neural network to predict the story point. Results: The experiments show that the proposed approach outperforms the state-of-the-art technique Deep-S which uses Recurrent Highway Networks. The pro-posed model has improved Mean Absolute Error (MAE) by an average of 16.6% and has improved Median Absolute Error (MdAE) by an average of 53%. Conclusion: An empirical evaluation shows that the proposed approach outperforms the previous work.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
216	Status indicators in software engineering group projects	A segment of studies on group structure and performance in software engineering (SE) project-based learning (PjBL) have focused on roles, including studies that use Belbin team roles and studies that address problematic roles such as social loafing. The present study focuses on the status, which is basically missing in SE PjBL studies, although relating to roles. The study investigates the aspects that students identified as indicators of rising or declining status in their project groups. The status theory was utilized as the framework that motivated the research and on which the results were reflected. An inductive qualitative content analysis was applied to learning reports in which students reflected on their statuses. The indicators of rising status included technical know-how, commitment, management responsibility, and idea ownership, while also group-level attributes such as a caring atmosphere and joint responsibility. The indicators of a declining status included aspects that appear as counterparts of rising status indicators, while also more refined aspects such as no one willing to be a leader or study background. The results are concluded to provide material for educating students about intra-group relations and promoting self-regulation for fruitful collaboration in groups. The authors believe that the results also initiate further PjBL research in which status theory can be utilized. (c) 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).	☑	☑	☐	☐	☐	☐	☐	☐
217	SOHCO: A Strategy for Constructing Efficient Teams	Software engineering is concerned with organizational issues, project management, and human behavior. In the process of constructing a work team, leadership faces the task of evaluating the talents and abilities of each professional and combining them into a cohesive unit that matches the profile of the project. This article describes the SOHCO technique, a strategy for forming work teams that calculates a score for each candidate based on the project profile, as well as a ranking for the work team whose desired objective most closely aligns with the project profile. This article presents exploratory research, an experiment, two real-world case studies, and a comparison of methods using WS and RW coefficients to assess ranking consistency and similarity. Results reveal that SOHCO makes constructing a work team more objective, reduces leadership effort, enhances the ability to evaluate new team arrangements, increases the probability of project success, and reduces training costs. As a limitation, the SOHCO method does not take sub-criteria weights into account.	☑	☑	☐	☐	☐	☐	☐	☐
218	Software engineering team project courses with industrial customers: Students' insights on challenges and lessons learned	Team project courses in software engineering allow students to apply their acquired disciplinary knowledge while developing essential skills needed to work in the software industry. This paper examines the challenges and lessons learned by students in two team project courses involving industrial customers. The first course involves small teams and less complex project, whereas the second course, has larger teams and more complex projects. Using thematic analysis, we analyzed 158 reports submitted by two cohorts of students across two successive team project courses. As per our findings most challenges and lessons learned pertain to soft skills, such as teamwork, working in remote and hybrid setting, and collaboration with industrial customers. The results show that challenges and lessons learned evolve as students progress to the second team project course, for example, managing changes and addressing individual skill gaps were more pronounced in the first project course, while students reported greater coordination, communication, and contribution issues in the second team project course. The alignment between the challenges faced and the lessons learned suggests that addressing challenges in teamwork, collaborating with industrial customers, and working in hybrid or remote settings helped students develop effective strategies to mitigate these challenges. This process offers a valuable learning experience for the students, enriching their professional growth.	☐	☐	☐	☐	☐	☐	☐	☐
219	Software Development Process Considerations in GNSS-Denied Navigation Project for Drones	This article will discuss the software development process utilised in the GNSS-denied navigation project targeted for drones. The process was implemented in an environment where software developers were oriented to developing a source code only and who showed great reluctance to follow any formal process. The mentioned process was a lightweight AGILE-based process that, by assumption, minimised developers' engagement in activities other than those related to the source code development. The process described in this paper was designed to support and reflect product quality characteristics like functional completeness and correctness, time behaviour, resource utilisation, analysability, modifiability and testability in an "implicit" way. It allowed the developers to achieve those characteristics in an "invisible" manner. This allowed us to achieve acceptable product quality without needing to engage experienced architects. The described process needs improvements and extensions to satisfy certification needs. However, for the prototyping or research phases, it may be a perfect solution for facilitating future product modifications and saving costs. The process improves product quality by underlying the value of transparency, performing specific actions and incorporating specific attributes related to team tasks. This research also focused on finding a metric related to process acceptance in a given environment, resulting in a three-degree scale showing whether the process needed improvements or was accepted by the environment. Such a metric is a new finding and the metric implies that there are some factors that stand behind any process, which determine whether it will be friendly to the environment (people with their habits, work style, personal traits, etc.) or not. However, investigating factors that determine the process acceptance and that are connected to the environment traits were beyond the scope of this research and will be considered in the future.	☑	☐	☐	☐	☐	☐	☑	☐
220	Software Design and Modeling Practices in an Online Software Engineering Course: The Learners' Perspective	Background. Global Software Engineering (GSE) education is an established practice in academia. Several methods and tools support communication and programming activities, but earlier development stages, such as software design and modeling practices, are less explored. Aim. The goal of this work is to analyze the learners' perspective during an online Software Engineering course. In particular, we focus on planning/organization activities and socio-technical challenges during the software design and modeling process. Method. We used a mixed-method approach to collect data from 30 undergraduate students enrolled in an online Software Engineering course. We combined questionnaires and interviews to analyze four GSE elements (i.e., communication practices, team collaboration, task allocation and distribution, and usage of collaboration tools). Moreover, we analyzed the socio-technical challenges faced by the teams. Results. Brainstorming is the most common practice used for planning software design and modeling activities. According to students, the usage of variant design notation is among the technical challenges. Despite the challenges, students would prefer to continue working in distributed teams. Conclusions. The result shares the lessons learned that can be helpful to build best practices for managing software design and modeling activities in GSE project-based courses. It includes the need to define standard architectural terminologies, standard list of collaboration tools, early identification of architectural artifact dependencies, frequent design reviews, and face-to-face kick-off meetings.	☑	☐	☐	☐	☐	☐	☐	☑
221	Snapshot Metrics Are Not Enough: Analyzing Software Repositories with Longitudinal Metrics	Software metrics capture information about software development processes and products. These metrics support decision-making, e.g., in team management or dependency selection. However, existing metrics tools measure only a snapshot of a software project. Little attention has been given to enabling engineers to reason about metric trends over time-longitudinal metrics that give insight about process, not just product. In this work, we present PRIME (PRocess MEtrics), a tool to compute and visualize process metrics. The currently-supported metrics include productivity, issue density, issue spoilage, and bus factor. We illustrate the value of longitudinal data and conclude with a research agenda. The tool's demo video can be watched at https://bit.ly/ase2022-prime. Source code can be found at https://github.com/SoftwareSystemsLaboratory/prime.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
222	Simulating a sustainable biorefinery process as autonomous learning tool for collaborative work in the chemical engineering degree	To improve the skills of the students in the Sustainable Refinery and Biorefinery subject of the Degree in Chemical Engineering, a learning tool has been implemented based on collaborative work, autonomy, and the use of information and communication technologies (ICT). This project includes the development of simulations that allow students to apply the learned concepts in a practical way, which were then assessed with the help of a specific rubric created for this purpose. Once the instrument was designed and implemented, its evaluation was carried out using a specific questionnaire. The results indicated that the use of simulation software in the classroom (with a 95 % acceptance rate among students) had a positive impact on the students, facilitating the comprehension of the content (scored a 3.9 out of 1-5 Likert scale), supporting their academic performance (close to 10 % average increase in the student qualifications), and fostering their professional profile in the field of Chemical Engineering (as supported by the rating of 4 out of 1-5 Likert scale). In addition, the constructive feedback provided by the students enabled the teaching team to identify areas for improvement and to consider future changes and updates to the pedagogical approach (as the organization of the seminars). Therefore, the use of different chemical process simulation software packages resulted in an essential strategy to develop the autonomous learning of the students and constituted a successful methodological tool, also preparing them to better meet the challenges of the industrial sector.	☑	☑	☐	☐	☐	☐	☐	☐
223	Scrum Watch: a tool for monitoring the performance of Scrum-based work teams	Agile Methods propose an approach for developing software based on an iterative and incremental life cycle model, in which needs and solutions evolve through collaboration between multi-functional and self-organized teams. As such, agile practices in work teams are gaining much momentum. To meet the demanding level of projects, agile software development also has to keep up with several challenges. In this context, software industry has chosen to use several tools to ease development and communication between different teams' members. However, these tools generate overwhelming volumes of data that hamper decision-making by project managers. To address this issue, we present Scrum Watch, a tool-based approach that focuses on generating, through cloud-based technologies, graphic elements and reports that assist project managers with information to support decision making. Results obtained from an undergraduate Systems Engineering course through a capstone project confirm the feasibility of the proposed approach, which exploits the benefits of the availability and visualization of process and product metrics.	☐	☐	☐	☐	☐	☐	☐	☐
224	Robotic-based invESTigation and mOnitoring of Ross sEa with the PROTEUS unmanned marine vehicle-The RESTORE project	In the framework of the Programma Nazionale di Ricerche in Antartide (PNRA) Robotic-based invESTigation and mOnitoring of Ross sEa (RESTORE) project, during the XXXVIII Italian Expedition in Antarctica (2022-2023), the Portable RObotic TEchnology for Underwater Surveys (PROTEUS) Unmanned Marine Vehicle (UMV) was used to carry out multidisciplinary scientific surveys in the Ross Sea. (PROTEUS) is an innovative UMV, developed by the Marine Robotics research group of CNR-INM, which is particularly suitable for operating in extreme environments such as the polar ones, thanks to its compact size, lightweight design, modularity, reconfigurability, and open hardware and software architectures. In the first period of the Italian Antarctic scientific expedition (PROTEUS) was used underwater, in the Remotely Operated Vehicle (ROV) configuration. In a second phase it was transformed into a USSV for operating semi-submersed, and in a third and final phase it was further transformed into a USV vehicle for operating on the sea surface. The versatility of PROTEUS allowed the onboard installation and integration of several sensors to acquire a comprehensive collection of bio-geo-chemical and physical parameters of the water column (conductivity, temperature, pressure, dissolved oxygen, turbidity and chlorophyll), and acoustic and video data of the ice and the seabed. This paper presents a detailed description of the design of PROTEUS, encompassing its modular architecture, propulsion and power supply systems, navigation, guidance and control system, and sensory payload. The field campaign described herein will demonstrate the exceptional adaptability of PROTEUS which enables novel methodologies for oceanic observations and data collection. PROTEUS is a non-conventional marine vehicle not only because of its distinctive design, but also due to its hybrid nature, which allows it to operate as four different marine robotic vehicles: ROV, AUV, USSV and USV. Moreover, the unique mechanics, electronics, and software architecture enable all robotic subsystems to work in concert, ensuring the coordinated functioning of the entire mechatronic device. Ultimately, PROTEUS incorporates an integrated data processing system capable of collecting and combining environmental and robotic telemetry. This renders PROTEUS a remarkable example of an innovative ocean engineering platform dedicated to data campaigns in Antarctica.	☑	☐	☐	☐	☐	☐	☑	☐
225	Return: Group Design Project for Pin-Point Landing Demonstrator using Drone Technologies	In a rapidly changing world, engineers must identify societal needs, solve problems with ingenuity, and create new knowledge. While conventional taught-based courses provide a solid foundation of knowledge, they often fall short in stimulating creative thinking, translating theoretical knowledge into real-world solutions, and fostering leadership and teamwork. This paper proposes a project-based learning course designed to solidify and amplify the technical knowledge gained in preceding taught-based modules while developing essential soft skills. The course covers various stages of designing a reusable launcher-like demonstrator, aiming to simulate take-off and pinpoint landing operations using drone technology. This involves identifying system requirements and designing both software and hardware components. Throughout the course, students enhance their ability to critically formulate, solve, and evaluate engineering problems. They gain and apply technical knowledge in all aspects of drone technology, including flight dynamics, control, navigation, guidance, situational awareness, and communication. Additionally, students explore key aspects of systems engineering practices, risk management, and time management. This paper details the problem design, course timeline, outcomes and key lessons learned from the course. Copyright (C) 2024 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)	☑	☐	☐	☐	☐	☐	☐	☑
226	RETRACTED: Intelligent Building Space Layout and Optimization Design Method Based on Biological Population Simulation (Retracted Article)	Smart buildings can store various types of data, analyze data at a faster speed, and quickly provide users with the most accurate and high-quality intelligent services through technical means. The main purpose is to meet the needs of users, to provide communication and a series of knowledge services for all professional participants, and to provide comfortable working resources and working environment for all professional participants to ensure the smooth implementation of the project and meet the construction engineering requirements. In this study, the spatial layout and optimization of intelligent buildings are carried out based on biological population simulation, and the following conclusions are drawn: (1) The emergence of multicore or multicore graphics converters reduces the cost and popularity of high-performance software platforms, and provides faster computing speeds for applications. A new platform and implementation method have been developed. Comparing marine biopopulation models relies on the i4Ocean platform. The i4Ocean platform is a computer platform for seascape information and virtual reality applications, including view management, production engine, and MVC core framework. (2) The swarm model simulates the movement of fish in a virtual ocean. Determine their behavioral changes based on determined behavioral conditions (e.g., in position, position, and velocity). As everyone in the group moves, the entire group is monitored for behavior. (3) Intelligent buildings involve many aspects of the social machine, such as building construction, which not only affects architecture, but also design. Smart construction helps improve quality, shorten production time, save resources, and control costs. It is an important part of the structural transformation of the supply side of the construction industry, and an important tool for the structural adjustment and modernization of the construction industry and greenhouses. (4) X2View is like the most popular configuration king, with integrated functions. Project software is placed on the X2View and sent to the door or touch screen for processing and monitoring. In the driver configuration interface, you can set the application frame time of the monitoring node and the real-time measurement time of the record value recording time frame.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
227	Relating team atmosphere and group dynamics to student software development teams' performance	Context: While the software engineering community (i.e., those involved with engineering software) is constantly in search of insights into team atmosphere and group dynamics and the way these issues impact team performance, little opportunities typically exist to explore this issue. Student projects offer an opportunity for us to understand these issues, and particularly if these students are on the verge of leaving university for post-study work and using similar practices to those used in industry.Objective: We explore a range of student software development projects' data and students' open-ended responses to five group dynamics categories: communication, time management, commitment, problem analysis and solving, and initiative and involvement.Method: We analyse both quantitative and qualitative data to study the variation in group dynamics across teams developing different software and how these variations correlated with team satisfaction. We also explore the group dynamics themes that evolve from students' open responses in relation to the five categories. Furthermore, we relate the prevalence of the themes to various software development performance metrics, before exploring the opportunity of predicting an optimum team dynamics.Results: We observe variations in the way different teams work, but higher performing teams also committed more to their projects. Various group dynamics themes were evident among functional teams, and specific patterns were more pronounced when teams were productive. Further, while there is no specific group dynamics pattern that predicts project success, successful teams were most organised and reflective.Conclusion: Competence may set the tone for positive group dynamics and team performance. Also, an achievement-driven orientation is as important as the soft skills and interpersonal aspects.	☐	☐	☐	☐	☐	☐	☐	☐
228	Refining Software Clustering: The Impact of Code Co-Changes on Architectural Reconstruction	Version control systems are essential for tracking and managing changes in software code. They also provide information about the relationships between software entities: when multiple entities (e.g., classes or interfaces) are frequently modified together, it may indicate an underlying connection between them. These code co-changes can be analyzed and filtered to uncover logical dependencies, which complement structural or lexical dependencies. While structural dependencies are extracted using code analysis techniques, logical dependencies can be extracted solely from version control system logs. This makes their extraction independent of the programming language. Our work investigates filtering techniques to eliminate co-change situations that are not meaningful as logical dependencies. The effectiveness of these filtering techniques is investigated by using the obtained logical dependencies in architectural reconstruction, a reverse engineering activity aimed at recovering a system's modular structure. This paper explores the use of logical dependencies as input for architectural reconstruction. The main goal is to investigate whether logical dependencies can complement or even fully replace structural dependencies in this process. We use clustering based on software dependencies to group related entities into modules. We conduct experiments on four open-source Java projects using three clustering algorithms (Louvain, Leiden, and DBSCAN) and two evaluation metrics (Modularization Quality and MoJoFM). We consider three different dependency configurations: 1) only structural dependencies, 2) only logical dependencies, and 3) their combination. Our goal is to assess which configuration performs best and to examine the advantages and limitations of using logical dependencies.	☑	☐	☐	☐	☐	☐	☑	☐
229	Reducing Requirements Ambiguity via Gamification: Comparison with Traditional Techniques	Requirements elicitation is one of the most significant activities of requirements engineering (RE) process. Poorly specified requirements can lead to a failed project. Various elicitation techniques are used to elicit requirements from the users and other stakeholders, each having its own pros and cons. Lack of user engagement, less user involvement, textual nature of the requirements, time taking process are some of the major problems that make it difficult to perform elicitation via traditional techniques. Moreover, these problems further create other challenges such as ambiguity, inconsistency, and incompleteness in requirements. Currently, researchers have focused on reducing ambiguity in requirements with the help of different techniques such as natural language processing techniques, requirement templates, and formal methods; however, these techniques work on reducing ambiguity during specification or from specified requirements. One of the "young' and exciting way of engaging users in requirements elicitation of a system is "Gamification', which helps in user engagement into the system. We intend to discover how gamification helps in reducing ambiguity by engaging stakeholders in an interactive manner. In this review study, we have reviewed traditional techniques used to detect and reduce requirements ambiguity. On the contrary, we have also presented the significance of gamification in requirements elicitation and the popular but effective game elements used in similar systems. Furthermore, this study highlights the significance of using gamification in requirements elicitation, which is beneficial to software development team as well as the users involved in the system.	☑	☐	☐	☑	☐	☐	☐	☐
230	Realising Meaningful Human Control Over Automated Driving Systems: A Multidisciplinary Approach	The paper presents a framework to realise "meaningful human control" over Automated Driving Systems. The framework is based on an original synthesis of the results of the multidisciplinary research project "Meaningful Human Control over Automated Driving Systems" lead by a team of engineers, philosophers, and psychologists at Delft University of the Technology from 2017 to 2021. Meaningful human control aims at protecting safety and reducing responsibility gaps. The framework is based on the core assumption that human persons and institutions, not hardware and software and their algorithms, should remain ultimately-though not necessarily directly-in control of, and thus morally responsible for, the potentially dangerous operation of driving in mixed traffic. We propose an Automated Driving System to be under meaningful human control if it behaves according to the relevant reasons of the relevant human actors (tracking), and that any potentially dangerous event can be related to a human actor (tracing). We operationalise the requirements for meaningful human control through multidisciplinary work in philosophy, behavioural psychology and traffic engineering. The tracking condition is operationalised via a proximal scale of reasons and the tracing condition via an evaluation cascade table. We review the implications and requirements for the behaviour and skills of human actors, in particular related to supervisory control and driver education. We show how the evaluation cascade table can be applied in concrete engineering use cases in combination with the definition of core components to expose deficiencies in traceability, thereby avoiding so-called responsibility gaps. Future research directions are proposed to expand the philosophical framework and use cases, supervisory control and driver education, real-world pilots and institutional embedding	☑	☐	☐	☐	☐	☐	☑	☐
231	Pros and Cons of running educational Hackathons in a genderneutral fashion	Our work builds knowledge on gender-neutral hackathons in educational context. Hackathon itself is an intensive projectbased event, requiring participants to make results in a short timeframe, which stimulates creative thinking, knowledge sharing, and overall learning. This has made hackathons popular in an educational context. As a hackathon event often implies software design, the introduction of hackathon practices would be especially beneficial for Software Engineering education. The question is how to attract a diverse student audience to enroll in these hackathon events and hackathon enhanced courses. Historically, hackathons attract fewer females than males and show gender balance issues. Even though there are positive examples of female-inclusive events, there is a risk that by focusing on women's interests, the hackathon designers could end up discouraging the male audience. It was found that men e.g., tend to reject feminine products and occupations as they experience social pressure regarding what is acceptable for them. Thus, it would be reasonable to focus on the hackathon's gender-neutrality rather than on female inclusivity. This research provides practices to implement hackathon courses gender-neutrally. We also discuss pros and cons of different approaches.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
232	Project-Sized Scaffolding for Software Engineering Courses	Students can often graduate with a degree in computer science without working with legacy code bases, yet when they join the workforce they will almost certainly work on an existing project with thousands, if not millions of lines of existing code. In order to give students a realistic experience without overwhelming them, we added scaffolding to an existing open source project and used it in our third year software engineering course. We asked students to complete a series of 5 tasks, from bug fixing to feature addition, with this scaffolded project. Our scaffolding consisted of enhanced documentation, demonstration videos, compilation videos, enhanced task descriptions, and hints for task completion. After running this course project we collected feedback via a survey (n=87) and a small focus group (n=7). We found that students appreciated the realistic experience, but that they recommend further scaffolding, especially within source code, to better balance between difficulty and learning.	☑	☑	☐	☐	☐	☐	☐	☐
233	Process automation in biotechnology engineering: reshaping and assessing active learning during disastrous events	Courses such as process automation (PA) can be difficult to grasp, even for engineering students. In addition to the inherent challenge of relating abstractions to real-life situations, we found that our students could not grasp basic concepts. After 5 years of work, we improved our students' performance in these fields. The solution involved redefining the course topics and closely monitoring their learning from the beginning of the course. The approach was enhanced with the support of specialized software and modifications to traditional active learning (AL) methods. Additionally, remote-mode AL was enabled by our institution in response to the 2017 earthquake and the 2019 pandemic. We investigated the difference delta = F-X between projects grades F and exams X via their medians. First, a test was applied to see whether the medians of delta were significantly different from 0. This result was true at 5% of level in 9 out of 12 classes, indicating improvement via F. Later, key groups 2015, 2017 and 2020 were similarly contrasted. Significant differences to zero were appreciated in groups 2015 vs. 2017, 2015 vs. 2020 (progress) but not in 2017 vs. 2020 may be due to those terrible events. Our novelty resulted statistically significant.	☐	☐	☐	☐	☐	☐	☐	☐
234	Practitioners' Perspective on Practices for Preventing Technical Debt Accumulation in Scientific Software Development	Scientific software development refers to a specific branch of software engineering that targets the development of scientific applications. Such applications are usually developed by non-expert software engineers (e.g., natural scientists, biologists, etc.) and pertain to special challenges. One such challenge (stemming from the lack of proper software engineering background) is the low structural quality of the end software-also known as Technical Debt-leading to long debugging and maintenance cycles. To contribute towards understanding the software engineering practices that are used in scientific software development, and investigating whether their application can lead to preventing structural quality decay (also known as Technical Debt prevention); in this study, we seek insights from professional scientific software developers, through a questionnaire-based empirical setup. The results of our work suggest that several practices (e.g., Reuse and Proper Testing) can prevent the introduction of Technical Debt in software development projects. On the other hand, other practices seem as either improper for TD prevention (e.g., Parallel / Distributed Programming), whereas others as non-applicable to the branch of scientific software development (e.g., Refactorings or Use of IDEs). The results of this study prove useful for the training plan of scientists before joining development teams, as well as for senior scientists that act as project managers in such projects.	☑	☐	☐	☐	☐	☐	☑	☐
235	Practicing Equity Diversity Inclusion (EDI) in Software Development Teams: A Systematic Literature Survey	Human factors in successful software projects have always been a critical element in software engineering, however, it has always been overshadowed by focusing more on technology and underlying processes. This work is inspired by the recent increasing interest from the software engineering research community in human factors and software development by leveraging and understanding some examples of human factors such as Equity, Diversity, and Inclusion (EDI) which were not given due research consideration earlier. We performed a systematic literature review (SLR) to review the state-of-the-art literature on practicing EDI in software development teams despite of country or culture. We found that evidence of comprehensive research about practicing EDI in software development teams is limited, the up-to-date majority focus is on the topic of diversity, whereas research on topics of practicing equity and inclusion in software development teams is sporadic. It is expected that investigating the impact of human factors in the context of EDI's triangle will generate new knowledge. This will allow software practitioners to understand the benefits of practicing EDI in managing software development teams as well as provide opportunities to incorporate them into the core development process activities. In the end, future research directions for EDI practices in software development teams are also identified.	☑	☑	☐	☐	☐	☐	☐	☐
236	Post-Pandemic Hybrid Work in Software Companies: Findings from an Industrial Case Study	Context. Software professionals learned from their experience during the pandemic that most of their work can be done remotely, and now software companies are expected to adopt hybrid work models to avoid the resignation of talented professionals who require more flexibility and work-life balance. However, hybrid work is a spectrum of flexible work arrangements, and currently, there are no well-established hybrid work configurations to be followed in the post-pandemic period. Goal. We investigated how software engineers are experiencing the post-pandemic hybrid work landscape, aiming to understand the factors that influence their choices between remote and in-office work. Method. We explored a large South American company by collecting quantitative and qualitative data from 545 software professionals who are currently navigating diverse hybrid work arrangements tailored to their individual and team requirements. Findings. Our study revealed an array of factors that significantly impact hybrid work within the software industry, including individual preferences, work-life balance, commute time, social interactions, productivity, and more. Team dynamics, project demands, client expectations, and organizational strategies also play an important role in shaping the complex landscape of hybrid work configurations in software engineering. Conclusions. In summary, the success of hybrid work models depends on balancing individual preferences, team dynamics, and organizational strategies. Our study demonstrated that, at present, there is no one-size-fits-all individuals approach to hybrid work in the software industry.	☑	☐	☐	☐	☐	☐	☑	☐
237	PoreVision: A Program for Enhancing Efficiency and Accuracy in SEM Pore Analyses of Gels and Other Porous Materials	Porous gels are frequently utilized as cell scaffolds in tissue engineering. Previous studies have highlighted the significance of scaffold pore size and pore orientation in influencing cell migration and differentiation. Moreover, there exists a considerable body of research focused on optimizing pore characteristics to enhance scaffold performance. However, current methods for numerical pore characterization typically involve expensive machines or manual size measurements using image manipulation software. In this project, our objective is to develop a user-friendly, versatile, and freely accessible software tool using Python scripting. This tool aims to streamline and objectify pore characterization, thereby accelerating research efforts and providing a standardized framework for researchers working with porous gels. Our group found that first-time users of PoreVision and ImageJ take similar amounts of time to use both programs; however, PoreVision is capable of handling larger datasets with reduced variability. Further, PoreVision users exhibited lower variability in area and orientation measurements compared to ImageJ, while perimeter variability was similar between the two. PoreVision showed higher variability in average measurements, likely due to its larger sample size and broader range of pore sizes, which may be missed in ImageJ's manual scanning approach. By facilitating quantitative analysis of pore size, shape, and orientation, our software tool will contribute to a more comprehensive understanding of scaffold properties and their impact on cellular behavior. Ultimately, we aim to aid researchers in the field of tissue engineering with a user-friendly tool that enhances the reproducibility and reliability of pore characterization analyses.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
238	PBL Student Assessment: Consistency of Different Evaluation Methods in a Computing Faculty	Contribution: This article analyzes student assessment within a computing faculty employing a full project-based learning (PBL) approach. Examining 2078 final grades across 60 classes and periods, the study reveals a significant correlation between graded self-studies, exams, and projects. This result contributes to understanding the reliability and independence of diverse evaluation methods, emphasizing their effectiveness in measuring students' learning within a PBL framework. Background: This study is motivated by the need to investigate diverse learning assessment methods within a PBL setting, aiming to establish context and underline the broad applicability of a comprehensive evaluation approach in computing education. Intended Outcomes: The study seeks to provide information about the interrelationships between different assessment methods within a PBL framework. The desired outcome is a comprehensive understanding applicable across various classes and periods, emphasizing the robustness of the assessment system. Application Design: Implementing a full PBL model in a computing faculty during a two-year period, the chosen approach integrates graded self-studies, exams, and projects to ensure a well-rounded evaluation system, capturing student proficiency across diverse computing concepts. This study involved 210 first-and second-year students, 162 men and 48 women, with a mean age of 21.7 +/- 2.7 years. Findings: Correlation analysis of 2078 final grades indicates consistent reliability among graded self-studies, exams, and projects. This result underscores the effectiveness of each assessment method in measuring learning within a PBL framework, highlighting the system's robust applicability in a computing educational environment.	☑	☑	☐	☐	☐	☐	☐	☐
239	Organizational Conflicts in the Adoption of Continuous Software Engineering	Software is a critical component of nearly every product or service. Improvements in software can lead to substantial competitive advantages. At the same time, software and surrounding engineering teams have become increasingly complex. The adoption of continuous integration and delivery is a recent trend to radically improve software release speed. However, its adoption is far from straightforward. Specifically, rethinking processes, organizational culture, ways of working, and business models require buy-in from diverse stakeholders that may have conflicting objectives. Such situations are explored by organizational conflict research. This paper reports on early lessons from an ongoing research project in continuous software engineering, specifically investigating adoption challenges from an organizational conflict perspective. We identify catalysts, symptoms, and outcomes of organizational conflicts hindering the adoption process. We conclude that predictable conflicts emerge when adopting continuous engineering. Engineers, managers, and other teams can proactively prepare for and allocate resources to resolve them. Proper analysis and management can help avoid wasted time, impeding processes, and frustration.	☑	☐	☐	☐	☐	☐	☑	☐
240	Optimization of Software Test Scheduling under Development of Modular Software Systems	Software testing and debugging is a crucial part of the software development process since defective software not only incurs customer dissatisfaction but also might incur legal issues. However, the managers of a software development company cannot arbitrarily prolong their software debugging period due to their software testing budget and opportunity in the market. Accordingly, in order to propose an advantageous testing project, the managers should be aware of the influence of the testing project on cost, quality, and time to make the best decision. In this study, a new software reliability growth model (SRGM) with consideration of the testing staff's learning effect is proposed to achieve better prediction. The methods of estimating the model's parameters and the symmetric confidence intervals are also proposed in the study. Moreover, in the past, most of the SRGMs focused on a single software system. However, in practice, some software systems were developed using modular-based system engineering approaches. Therefore, traditional software testing work can be changed to multiple modular testing work in this scenario. Therefore, the manager can use this to dispatch multiple staff groups to perform the individual testing work simultaneously. The study proposes two mathematical programming models to handle the scheduling of modular testing work. Additionally, the design of a computerized decision support system is also proposed in the study for the application in practice.	☑	☐	☐	☐	☐	☐	☑	☐
241	Optimization and Impact Assessment of Excavation Sequence around Subway Stations from the Perspective of Sustainable Urban Development	The construction of group foundation pits near subway stations often leads to environmental pollution, thereby causing certain damage to urban ecology. By optimizing the excavation sequence of group foundation pits, the adverse effects on surrounding underground structures and soil during excavation can be effectively mitigated, contributing to the sustainable development of cities. Taking a group foundation pit project in Changzhou as an example, this study utilized the finite element software PLAXIS 3D to simulate various working conditions under different excavation sequences, comparing the deformation of the subway station, shield tunnel, and surrounding soil. The results show that, influenced by the excavation of group foundation pits, the difference between maximum deformation and minimum deformation of shield tunnel is 25.85%, and the difference between the maximum deformation and minimum deformation of the subway envelope is 19.44%. The subway envelope is least affected by the change in excavation sequence. Both the displacement of the subway station and the surrounding soil exhibit a significant "cumulative effect", with displacement changes closely related to the distance from the pit to the station and the ground, as well as the amount of soil unloaded in each excavation. Therefore, it is advisable to adhere to the principle of "far before near, shallow before deep, small before large" during excavation, which facilitates the coordinated development of urban infrastructure construction and the urban ecological environment, providing valuable reference and guidance for the sustainable development of cities.	☑	☐	☐	☐	☐	☐	☑	☐
242	Open Inventive Design Method (OIDM-Triz) Approach for the Modeling of Complex Systems and the Resolution of Multidisciplinary Contradictions. Application to the Exploration of Innovative Solutions to Deal with the Climate Change Impacts	This paper aims to propose an approach for Open IDM-Triz oriented analysis for complex systems. We propose an approach to model a complex system involving multiple points of view. In this way, a complex system is modelled by a meta-graph that is built using partial knowledge graphs based on the viewpoints of the different stakeholders and users around the complex system. This work is also a contribution for resolving non-conventional pluridisciplinary contradictions by extending the limits of classic Triz patents databases which is mainly focused on technical aspects. Thereby, we propose an approach to extend IDM-Triz solutions concepts exploration area by using web-scraping and AI reasoning to search for similar contradictions. These approaches are implemented in a TrizAlerts software that scans various databases and websites to find out ideas matching given contradictions. Our Open IDM-Triz approach is applied in the Interreg Clim'Ability Design project, to analyze the climate change impacts on the activities that usually manifest themselves in multiple ways. By applying the Open IDM-Triz approach to the "Low Water" issue, we are moving away from the classical application domain of Triz because of the context extended to a multi-scalar territory corresponding to a complex system of actors impacted by climate change. The study is carried out by a multidisciplinary team made up of researchers from engineering and social sciences backed by a network of companies involved in logistics on the Rhine (institutional, shippers, operators, etc.).	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
243	OntoTrace: A Tool for Supporting Trace Generation in Software Development by Using Ontology-Based Automatic Reasoning	Traceability in software development has gained interest due to its software maintainability and quality assurance benefits. Artifacts such as code, requirements, mockups, test cases, among others, are feasible trace sources/targets during the software development process. Existing scientific approaches support tasks like identifying untraced artifacts, establishing new traces, and validating existing traces. However, most approaches require input existing traceability data or are restricted to a certain application domain hindering their practical application. This contemporary challenge in information systems engineering calls for novel traceability solutions. In this paper, we present OntoTrace: a tool for supporting traceability tasks in software development projects by using ontology-based automatic reasoning. OntoTrace allows software development teams for inferring traceability-related data such as i) which are the traceable source/target artifacts; ii) which artifacts are not yet traced; and iii) given a specific artifact, which are the possible traces between it and other artifacts. We demonstrate how OntoTrace works in the context of the Swiss startup LogicFlow AG, supporting the traceability between functional/non-functional requirements and user interface test cases. We conclude the paper by reflecting on the experience from applying the approach in practice, and we draw on future challenges and next research endeavors.	☑	☐	☐	☐	☐	☐	☑	☐
244	On the relationship between similar requirements and similar software A case study in the railway domain	Recommender systems for requirements are typically built on the assumption that similar requirements can be used as proxies to retrieve similar software. When a stakeholder proposes a new requirement, natural language processing (NLP)-based similarity metrics can be exploited to retrieve existing requirements, and in turn, identify previously developed code. Several NLP approaches for similarity computation between requirements are available. However, there is little empirical evidence on their effectiveness for code retrieval. This study compares different NLP approaches, from lexical ones to semantic, deep-learning techniques, and correlates the similarity among requirements with the similarity of their associated software. The evaluation is conducted on real-world requirements from two industrial projects from a railway company. Specifically, the most similar pairs of requirements across two industrial projects are automatically identified using six language models. Then, the trace links between requirements and software are used to identify the software pairs associated with each requirements pair. The software similarity between pairs is then automatically computed with JPLag. Finally, the correlation between requirements similarity and software similarity is evaluated to see which language model shows the highest correlation and is thus more appropriate for code retrieval. In addition, we perform a focus group with members of the company to collect qualitative data. Results show a moderately positive correlation between requirements similarity and software similarity, with the pre-trained deep learning-based BERT language model with preprocessing outperforming the other models. Practitioners confirm that requirements similarity is generally regarded as a proxy for software similarity. However, they also highlight that additional aspect comes into play when deciding software reuse, e.g., domain/project knowledge, information coming from test cases, and trace links. Our work is among the first ones to explore the relationship between requirements and software similarity from a quantitative and qualitative standpoint. This can be useful not only in recommender systems but also in other requirements engineering tasks in which similarity computation is relevant, such as tracing and change impact analysis.	☑	☐	☐	☐	☐	☐	☑	☐
245	On the Impact of Grading on Teamwork Quality in a Software Engineering Capstone Course	Every semester, we deliver a capstone course on software engineering where students undertake a real-world project in three iterations. By the end of each iteration, students are graded in several dimensions: software quality, project management, and peer assessment. The latter is the only grade assigned individually; therefore, students who are penalized by their teams (e.g., for being perceived as low contributors to the team effort) are not severely affected. This results in little incentive for improvement, which potentially jeopardizes the overall quality of the project outcome. Envisaging to promote team cohesion, we devised a new grading rule: if the peer assessment of a student is lower than a threshold, that would be their final grade in the iteration. This paper reports the results of studying the effectiveness of the proposed rule. We recorded peer assessments over six consecutive semesters: (1) the first three as the baseline measure; (2) the semester where we introduced the new grading rule; and (3) the following two semesters, as a contrast. When the rule was first introduced, peer assessments resulted low and heavily spread at the beginning, but they consistently improved toward the end of the semester. When the instructional team already trusted the rule and explicitly emphasized its application, peer assessments consistently grew along the semester but resulted in fewer outliers. The study results show that exposing peer assessments earlier on helps promote team reflection. They also made evident the positive impact of teamwork for producing quality products in a software engineering capstone course.	☑	☑	☐	☐	☐	☐	☐	☐
246	On the Application of Agile Project Management Techniques, V-Model and Recent Software Tools in Postgraduate Theses Supervision	Due to the nature of most postgraduate theses in control engineering and their similarities to industrial and software engineering projects, invoking novel project control techniques could be effective. In recent decades, agile techniques have attracted popularity thanks to their attributes in delivering successful projects. Hence exploiting those methods in education and thesis supervision of engineering topics can facilitate the process. On the other hand, because of the limitations imposed by the CoVid19 pandemic, the integration of well-established online tools in collaborative education is noteworthy. This paper proposes an application of the agile project management method for the supervision of postgraduate students' theses in the general field of engineering. The study extends a Scrum technique combined with approved systems engineering and team working tools such as Jira Software, Microsoft Teams, and Git version control (Github website). A custom designed V-model to nail an outstanding thesis is presented. The overall blended method is beneficial to provide feedback and self-assessment aid for the students and the supervisors. Employing this technique has shown promising progress in easing the supervision of students whilst helping them to manage their projects. Copyright (C) 2022 The Authors.	☑	☐	☐	☐	☐	☐	☑	☐
247	Numerical Investigations of Pile Group Foundations under Different Pile Length Conditions	At present, group pile foundations with the same length of pile base are used basically in large-scale slope group pile foundation projects. Therefore, pile group foundations with piles of different lengths have a certain research value. Based on the actual working condition of a bridge group pile foundation, a similar model is established, which is imported into the FLAC3D 6.0 finite element software package together with the processed relevant data, and the bearing performance of the cap-group pile foundation under the joint action of axial uniform load and landslide thrust is studied. The study shows: under the same bearing conditions, the settlements of group pile foundations with the same pile length and different pile lengths are similar, and the settlements of the rear row of piles is significantly higher than those of the front row of piles; the settlement of the cap platform in the area without backfill soil is different from that in the area with backfill; the front row of piles has some negative displacement within the range of 10 m below the equivalent sliding surface, and the displacement of the pile body from the back row of piles to the front row of piles increases linearly; the maximum bending moment of the foundation pile is at the position of the gravel soil layer, and as the load changes, the position of the maximum bending moment point will also change; the plastic zone of the uppermost gravel soil layer in the slope model has the tendency of penetration, but it is truncated by the group of piles, and the factor of safety is 2.4 in the case of 100 KN axial uniform load, this structure tends to be stabilized, and the factor of safety decreases with the increase in the load. The analysis of the bearing characteristics of group piles under horizontal and vertical loads and its related conclusions can be used as a reference for related engineering design.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL				EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
248	Motivating members' involvement to effectually conduct collaborative software process tailoring	Contemporary business and software environments are highly competitive and rapidly evolving, resulting in software projects that are highly customized and changeable during development. Therefore, software process tailoring (SPT) is important as software teams conduct SPT to adjust shared development processes and evolve the project to better meet unique and dynamic needs. SPT is a special type of teamwork in which members' active participation and critical input are necessary for understanding and synthesizing various business and technical concerns that may be divergent and conflictual and then jointly identifying an integrated tailoring solution. In this context, this study examines members' decisive and critical involvement in SPT and adopts a motivational perspective to explore how motivation can facilitate SPT performance. Specifically, we use empowerment theory to develop a model to theorize and examine how psychological empowerment (PE) in terms of meaningfulness, autonomy, potency, and impact motivates software teams to efficiently and effectively conduct SPT. The model also considers the power distance (PD) to understand how it functions in team-based critical thinking and decisional processes to energize team members' participative effort. The investigation surveyed 102 software development teams and used partial least squares (PLS) to analyze the data. The results show that PE in terms of the four components has various influences on SPT performance and that PD has nonsignificant moderating effects. This study contributes to the software engineering literature by uncovering the contextual mechanism underlying the relationship between PE and PD in SPT. The limitations and possible extensions of this study are also outlined for future research.	☑	☐	☐	☐	☐	☐	☑	☐
249	MONITORING AND ANALYSIS OF STUDENTS' PERFORMANCE DURING SOFTWARE DEVELOPMENT	This study presents a comprehensive approach to monitoring and analyzing students' performance in the context of software development projects. The primary focus is on the development of novel performance metrics that effectively capture the level of control required from the Team Leader and the quality of work assessed by the QA Engineer. The introduced metrics, namely Relative Control Degree and Relative Quality Assurance Degree, offer a robust framework for evaluating and understanding students' ' performance across project, company, and industry levels. This study presents an innovative application that calculates the defined performance metrics, allowing for real- time monitoring of students' performance as a Developer. This practical tool offers means to track and evaluate student progress throughout the project lifecycle, enabling timely feedback and intervention as needed. The authors provide recommendations for interpreting the results and utilizing the proposed performance metrics. The utilization of the developed performance metrics holds significant implications for project management and student development. By implementing the recommended practices, educational institutions and industry professionals can enhance the effectiveness of software development projects, improve student learning outcomes, and promote a culture of continuous improvement and innovation. The following methods, models, and approaches were used to solve the task of improving monitoring and analyzing the progress of students in the process of participating in software development projects: System and Problem Approaches, Cluster Analytical and Monitoring Models of Software Development, Expert Survey, Consistency Index, Consistency Ratio, the Fundamental Scale of Absolute Numbers by T. L. Saaty, and others. The proposed performance metrics, supported by recommendations and a dedicated application, empower project stakeholders to optimize control, quality assurance, and student development.	☐	☐	☐	☐	☐	☐	☐	☐
250	Loss Assessment of Non-Ductile RC Buildings Accounting for Bidirectional Ground Motion	The ongoing seismic events in the world highlight the significant vulnerability of the existing building stock. In particular, in Italy, several structures have been designed without seismic criteria and showed significant damage after recent earthquakes, causing many casualties and great economic losses. Earthquake impact assessment has become one of the most investigated topics by several research groups and within national and international projects. In Italy, for example, the National Department of Civil Protection has allocated several economic funds within the ReLUIS project to define methodologies able to support the decision-making process for seismic prevention activities and risk reduction. This work aims to assess economic losses using the methodology proposed by FEMA-P58, according to the Performance-Based Earthquake Engineering method, evaluating the influence of bi-directional ground motion on loss estimates. The analyzed case study is a three-dimensional nonlinear frame characterized by one weak and one strong seismic directions and modeled using OpenSees software. The model considers the joint deformability and the flexural and shear damaging behavior of beams and columns. Thirty time-history analyses were performed considering bi-directional ground motions applied in the two main directions of the building and applying the higher spectral acceleration firstly in the weaker direction and then in the stronger one. Eight increasing hazard levels were selected for the nonlinear time-history analyses, and the corresponding structural analysis results, in terms of interstory drift ratio and peak floor acceleration, were implemented for the loss assessment. The results of the loss assessment are then discussed in terms of expected annual loss and loss hazard curve.	☑	☑	☐	☐	☐	☐	☐	☐
251	lifex-fiber: an open tool for myofibers generation in cardiac computational models	Background: Modeling the whole cardiac function involves the solution of several complex multi-physics and multi-scale models that are highly computationally demanding, which call for simpler yet accurate, high-performance computational tools. Despite the efforts made by several research groups, no software for whole-heart fully coupled cardiac simulations in the scientific community has reached full maturity yet.Results: In this work we present life(x)-fiber, an innovative tool for the generation of myocardial fibers based on Laplace-Dirichlet Rule-Based Methods, which are the essential building blocks for modeling the electrophysiological, mechanical and electromechanical cardiac function, from single-chamber to whole-heart simulations. life(x)-fiber is the first publicly released module for cardiac simulations based on life(x), an open-source, high-performance Finite Element solver for multi-physics, multi-scale and multi-domain problems developed in the framework of the iHEART project, which aims at making in silico experiments easily reproducible and accessible to a wide community of users, including those with a background in medicine or bio-engineering.Conclusions: The tool presented in this document is intended to provide the scientific community with a computational tool that incorporates general state of the art models and solvers for simulating the cardiac function within a high-performance framework that exposes a user-and developer-friendly interface. This report comes with an extensive technical and mathematical documentation to welcome new users to the core structure of life(x)-fiber and to provide them with a possible approach to include the generated cardiac fibers into more sophisticated computational pipelines. In the near future, more modules will be successively published either as pre-compiled binaries for x86-64 Linux systems or as open source software.	☑	☐	☐	☐	☐	☐	☑	☐
252	Learning to Work From Home: How Novice and Experienced Sofware Professionals Compare Online and In-person Collaboration	Industry-sponsored capstone experiences that involve developing software solutions create valuable opportunities for students to engage in teamwork and learn important career skills just prior to joining the software workforce. The collaboration and communication in capstone projects, which typically happen in person, were forced to move online during the COVID-19 pandemic. We leveraged the transition to conduct a two-stage mixed-methods study that compared the in-person and online capstone experiences of students (novice software developers) and project sponsors (experienced industry professionals) to understand how novices engage in online collaboration. We uncovered that novices fnd online collaboration more challenging compared to experienced professionals. The challenges faced by the novices and the misalignment with experienced professionals point to a number of avenues for training novice software professionals on effective engagement in geographically distributed collaboration, which is increasingly the norm in the software industry.	☑	☐	☐	☐	☐	☐	☐	☑
253	Learning Soft Skills through Distributed Software Development	The software industry needs universities to train developers to have besides the technical skills, also strong soft skills to collaborate in globally distributed software development projects. To develop these soft skills, we organized a distributed online software development project course, during which student Scrum teams of 5-8 members from five Belarusian universities worked in industrial projects for Danish customers. The course aimed to 1) teach students the Scrum framework and soft skills, such as teamwork and communication with international customers; and 2) to give Belarusian teachers ideas for organizing similar courses in the future. Based on 20 post-course semi-structured interviews with students and stakeholders, and the analysis of 24 student learning diaries we studied the learning outcomes and challenges related to soft skills. The main reported learning outcomes were: communication, methodical use of Scrum, problem solving, organizational/planning skills, teamwork, interpersonal skills, and time management.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
254	Lean Software Startup Practices and Software Engineering Education	In the modern economy, software drives innovation and economic growth. Studies show how software increasingly influences all industry sectors. Over the past 5 decades, software engineering has also changed significantly to advance the development of various types and scales of software products. Software engineering education plays an essential role in apprising students of software technologies, processes, and practices popular in industries. Furthermore, approaches to teaching software engineering are becoming more interdisciplinary and team-centered, comparable to startup contexts. In this PhD work, I want to answer the following research questions: (1) To what extent are software engineering trends present in software engineering education research? (2) What set of common software engineering practices employed in lean software startups is transferable to the software engineering education context? (3) What is the impact of lean startup practices on software engineering students and curricula? I utilize (1) a literature review, (2) mixed-methods approaches in gathering empirical evidence, and (3) design-based research. In the first phase of the research, I pinpoint the relevance of the lean startup in software engineering education through an extensive literature review. I gather empirical evidence on lean startup practices and assess their potential transferability to software engineering education during the second research phase. I demonstrate that the lean startup is an emerging trend in software engineering education research. I demonstrate that students can acquire soft, hard, and project management skills in a more realistic context in the introduction of the growth phase of lean startup practices throughout external course activities. I expect software engineering curricula to benefit from the model and framework that I propose and validate, thus facilitating lean startup practice transfer to software engineering curricula.	☑	☑	☐	☐	☐	☐	☐	☐
255	"It's Not Exactly Meant to Be Realistic": Student Perspectives on the Role of Ethics In Computing Group Projects	In computing education, group projects are often seen as an opportunity for students to gain experiences similar to what they will face in the workplace. As such, it is a pressing area for ethics education: practice incorporating ethics into project-based courses might provide encouragement and scaffolding for students to consider ethics when working on "real" technologies. In this work, we provide a preliminary look at the state of ethics education in project-based computing courses via semi-structured focus groups. These focus groups revealed that while not completely ignored, ethics had a very small role in our participants' group projects, especially in software engineering courses. Furthermore, participants generally agreed that while group projects can be useful learning opportunities, they are not realistic, and this impacts their willingness to consider ethical implications while working on projects. We compiled participant feedback and ideas on how to improve ethics education in computing group projects, and present a synthesis on how this feedback can be implemented in the classroom. We also emphasize that part of simulating "real" tech work should include acknowledgment of larger structural problems and ideally training and practice in communication and argumentation.	☑	☐	☐	☐	☐	☐	☐	☑
256	Issues in the Adoption of the Scaled Agile Framework	Agile methods were originally introduced for small sized, colocated teams. Their successful products immediately brought up the issue of adapting the methods also for large and distributed organizations engaged in projects to build major, complex products. Currently the most popular multi-teams agile method is the Scaled Agile Framework (SAFe) which, however, is subject to criticism: it appears to be quite demanding and expensive in terms of human resource and project management practices. Moreover, SAFe allegedly goes against some of the principles of agility. This research attempts to gather a deeper understanding of the matter first reviewing and analysing the studies published on this topic via a multivocal literature review and then with an extended empirical investigation on the matters that appear most controversial via the direct analysis of the work of 25 respondents from 17 different companies located in eight countries. Thus, the originality of this research is in the systemic assessment of the "level of flexibility" of SAFe, highlighting the challenges of adopting this framework as it relates to decision making, structure, and the technical and managerial competencies of the company. The results show that SAFe can be an effective and adequate approach if the company is ready to invest a significant effort and resources into it both in the form of providing time for SAFe to be properly absorbed and specific training for individuals.	☑	☐	☐	☐	☐	☐	☑	☐
257	Is Modeling Access Control Worth It?	Implementing access control policies is an error-prone task that can have severe consequences for the security of software applications. Model-driven approaches have been proposed in the literature and associated tools have been developed with the goal of reducing the complexity of this task and helping developers to produce secure software efficiently. Nevertheless, there is a lack of empirical data supporting the advantages of model-driven security approaches over code-centric approaches, which are the de-facto industry standard for software development. In this work, we compare the result of implementing the same functional and security requirements by multiple developer groups in the context of a security engineering graduate course. We thereby obtain evidence on the security and efficiency of a tool-based modeldriven approach to security from the literature compared to a direct implementation in a well-known, modern web-development framework. For example, the projects using model-driven development pass up to 50% more security tests on average with less development effort. Also, we observe that models are twice as concise as manual implementations, which improves system maintainability.	☑	☐	☐	☐	☐	☐	☑	☐
258	Integrating Small-scale Autonomous Vehicles in CS Education: An Experience Report	Teaching software systems engineering is neither effective, nor inspiring if students cannot practice the conveyed theory. We report on a hands-on approach that closes the gap by means of off-the-shelf and in-house realistic miniature models of autonomous vehicles. It is innovative that the infrastructure extends beyond the widely-used solutions, to digitally controlled cars and trains, with open hardware and extendable sensory/actuating capabilities (cameras, opto-sensors, proximity sensors, LEDs, displays) operating in realistic mini-environments, including line-marked roads, railways, traffic-lights and -signs, tunnels and railroad switches. For many years already, we have been using these vehicles to structurally teach physical computing (300 undergraduates yearly) and systems testing (12 graduates yearly), and for incidental individual research projects. Recently, we started two initiatives that stimulate a more dynamic know-how transfer across generations. The first is a better scalable undergraduate capstone project, where groups of students work on a 'hot' research topic in autonomous driving. The second motivates student teams to go the extra mile in an international intelligent car race competition. Evaluations showed that although unusual for a non-engineering curriculum, 'playing' with autonomous vehicles is an excellent strategy to discover the interaction of software with hardware and environment, to better consolidate existing knowledge on programming and testing, and to explore new fields, such as AI-based computer vision, navigation and safety, adding in all cases more fun and motivation. We share the design of the scaffolding and teaching initiatives, together with lessons learned which will hopefully inspire other educators in shaping engaging and future-proof CS curricula.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
259	Innovation meets institutions: AI and the Finnish construction ecosystem	Artificial intelligence (AI), machine learning (ML), and deep learning (DL) are technologies that have recently transformed many industries. The construction industry has traditionally been a laggard industry in terms of digital-technology adoption. When leading firms in this industry have experimented with these technologies, many of these experiments have met resistance. In this paper we take an institutional lens to study why and particular social structures appears to have contributed to the resistance and paucity of success stories. Within institutional research, we focus on research with traces to cognitive science and psychology. We have carried out a qualitative embedded multiple-case study on resistance to new technologies and how to overcome such resistance. The study involves four use cases in the Finnish construction industry: (1) automation of a material-product subcontractor's production planning; (2) business-model innovation by contractor on how to best work across multiple construction sites at once; (3) machine learning and automation of documentation by a software firm; and (4) promotion of a vision of information sharing across organizations by the above software firm. Based on within and cross-case analyses, preliminary empirical findings are that AI, ML and DL have in the Finnish construction industry challenged institutionalized forms of organizing and workflow established long since in the industry and, until about the time of this piece of research, taken for granted. Resistance was nonetheless beginning to be overcome at the time of writing this piece of research with small-group interaction across firms- such as those in this study- in the industry ecosystem. Human-human mediation and face-to-face encounters were building trust in and across the organizations. The implication for practice and policy is that business transformation will not quickly and autonomously transform into "impersonal" or machine-machine exchange but, before that, requires human-human mediation. "In the long-term, AI and analytics have boundless potential use cases in E&C [i.e. engineering and construction]. Machine learning is gaining some momentum as an overarching use case (that is, one applicable to the entire construction life cycle, from preconstruction through O&M 8i.e. operations and management), particularly in reality capture (for example, in conjunction with computer vision) as well as for comparison of in situ field conditions with plans (for example, supporting twin models). Indeed, by applying machine learning to an ongoing project, schedules could be optimized to sequence tasks and hit target deadlines, and divergences from blueprints could be caught closer to real time and corrected using a variety of predetermined potential scenarios." [1]	☑	☐	☐	☐	☐	☐	☑	☐
260	Industry Experiences with Large-Scale Refactoring	Software refactoring plays an important role in software engineering. Developers often turn to refactoring when they want to restructure software to improve its quality without changing its external behavior. Small-scale (floss) refactoring is common in industry and is often performed by a single developer in short sessions, even though developers do much of this work manually instead of using refactoring tools. However, some refactoring efforts are much larger in scale, requiring entire teams and months or years of effort, and the role of tools in these efforts is not as well studied. In this paper, we report on a survey we conducted with developers to understand large-scale refactoring and its tool support needs. Our results from 107 industry developers demonstrate that projects commonly go through multiple large-scale refactorings, each of which requires considerable effort. Our study finds that developers use several categories of tools to support large-scale refactoring and rely more heavily on general-purpose tools like IDEs than on tools designed specifically to support refactoring. Tool support varies across the different activities, with some particularly challenging activities seeing little use of tools in practice. Furthermore, our analysis suggests significant impact is possible through advances in tool support for comprehension and testing, as well as through support for the needs of business stakeholders.	☑	☑	☐	☐	☐	☐	☐	☐
261	Improving Software Engineering Teamwork with Structured Feedback	Teamwork is a key learning outcome for our course, Foundations of Software Engineering. However, conflicts are inevitable in teams, and if students cannot resolve conflicts, this can lead to decreased satisfaction for everyone on the team. In this experience report, we present our approach to help students deal with conflicts that can occur in team projects. Working with faculty from the School of Business, following best practices of organizational behaviour research, we instructed students on how to provide high quality peer feedback, and designed activities where students provided feedback to each other following these principles. After this intervention, we compared the results of our teamwork survey with the results from the previous semester. We saw a meaningful drop in teamwork problems directly after the intervention, and the effect persisted for the rest of the semester.	☑	☑	☐	☐	☐	☐	☐	☐
262	Impact of Servant Leadership on Project Success Through Mediating Role of Team Motivation and Effectiveness: A Case of Software Industry	Project management in today's intensely competitive environment has compelled organizations to adopt project management approach for better business results. Therefore, current research study aims to investigate the impact of servant leadership on project success with the mediating role of team motivation and team effectiveness in the software industry. A field survey was conducted, using a questionnaire as a survey tool. Data were collected from 219 respondents who have been working as team members of software development projects. The statistical results were obtained using the SPSS Process macro. The results show project managers need to exhibit a servant leadership style due to its strong influence on project success, albeit through team motivation and effectiveness. The findings from this study contribute to the field of leadership and project management along with the field of information systems and software engineering.	☑	☐	☐	☐	☐	☐	☑	☐
263	Impact of Emotional Intelligence on Leadership and Team Dynamics in Agile Software Engineering Projects	Agile software engineering emphasizes collaboration, adaptability, and team cohesion, where Leadership and interpersonal dynamics are critical for project success. Emotional Intelligence (EI), the ability to recognize, understand, and manage emotions, has been theorized to enhance communication, conflict resolution, and team performance in Agile environments. However, empirical evidence on its direct and mediated impacts remains underexplored. This study examines how EI influences leadership effectiveness and team dynamics in Agile software projects, focusing on quantifying these relationships to inform actionable strategies. A quantitative approach was employed, utilizing structured surveys from 51 Agile professionals (Scrum Masters, developers, and product owners) actively engaged in frameworks such as Scrum, Kanban, and Extreme Programming (XP). Validated instruments-the Emotional Intelligence Scale (EIS), Multifactor Leadership Questionnaire (MLQ), and Team Climate Inventory (TCI) measured EI competencies, leadership behaviors, and team dynamics. Data were analyzed using correlation, regression, and mediation analyses. Results revealed significant positive correlations between EI and leadership effectiveness ($r=0.504$, $p<0.01$) and EI and team dynamics ($r=0.434$, $p<0.01$). Mediation analysis demonstrated that leadership effectiveness fully mediates the relationship between EI and team outcomes (beta =0.290, $p<0.01$), highlighting the pivotal role of emotionally intelligent leaders in fostering trust and adaptability. Teams led by individuals with high EI reported a 20% increase in trust and marked improvements in collaboration and decision-making. These findings underscore EI as a critical catalyst for Agile success. Organizations are urged to prioritize EI development through targeted leadership training programs and integrate Digital Emotional Intelligence (DEQ) tools (Zoom and Microsoft Teams) to bridge emotional gaps in distributed teams. This study provides actionable insights for optimizing team performance, conflict resolution, and project outcomes in fast-paced software engineering environments, advocating for a human-centric approach to Agile methodologies.	☑	☑	☐	☐	☐	☐	☐	☐
264	How many pomodoros do professional engineers need to complete a microtask of programming? -Two Industrial Case Studies of Freelancers and Employees using Microtask Programming-	Microtask programming enables software engineers such as freelancers and part-time employees to contribute to software projects even when they can not spend much time on them. It decomposes software design into small, self-contained specifications. The decomposed specifications enable them to complete implementation and review task in a short time. In this paper, we empirically investigate the time required for software engineers to complete microtasks in an industrial setting and explore their perceptions of microtask programming by investigating two industrial projects using it. The projects were carried out in different companies and differed in the employment of the engineers. One contracted 9 freelancers, and the other asked for 8 part-time contributions from employees at work on other projects. We conducted a survey and a focus group with the engineers. Based on the development data of the case studies, we found that almost all microtasks were completed in less than four pomodoro repetitions, namely about two hours in the pomodoro technique. These data shows that engineers who cannot work full-time on a project can undertake microtasks if they can spare one-third of their work day. We also examine how engineers who are employees experience microtask programming similarly and differently from freelancers.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
265	Hints on Designing and Running Project-based Exams for a Software Engineering Course	In this paper, we share our experience in designing and running a project-based exam for a Software Engineering course at the undergraduate level. We underline the teaching objectives and content of the course which aim to prepare students for the industrial environment, e.g., various design perspectives on the project, software quality assurance and evaluation, tools for development and software quality assessment, project management and team work. We present our project-based exam and summarize hints on its design, organization, and evaluation. Project examples are also introduced.	☑	☑	☐	☐	☐	☐	☐	☐
266	Guidelines for the development of a critical software under emergency	Context: During the first wave of the COVID-19 pandemic, an international and heterogeneous team of scientists collaborated on a social project to produce a mechanical ventilator for intensive care units (MVM). MVM has been conceived to be produced and used also in poor countries: it is open-source, no patents, cheap, and can be produced with materials that are easy to retrieve.Objective: The objective of this work is to extract from the experience of the MVM development and software certification a set of lessons learned and then guidelines that can help developers to produce safety-critical devices in similar emergency situations. Method: We conducted a case study. We had full access to source code, comments on code, change requests, test reports, every deliverable (60 in total) produced for the software certification (safety concepts, requirements specifications, architecture and design, testing activities, etc.), notes, whiteboard sketches, emails, etc. We validated both lessons learned and guidelines with experts. Findings: We contribute a set of validated lessons learned and a set of validated guidelines, together with a discussion of benefits and risks of each guideline.Conclusion: In this work we share our experience in certifying software for healthcare devices produced under emergency, i.e. with strict and pressing time constraints and with the difficulty of establishing a heterogeneous development team made of volunteers. We believe that the guidelines will help engineers during the development of critical software under emergency.	☑	☐	☐	☐	☐	☐	☑	☐
267	Guidelines for collaborative development of sustainable data treatment software	Software development for data reduction and analysis at large research facilities is increasingly professionalized, and internationally coordinated. To foster software quality and sustainability, and to facilitate collaboration, representatives from software groups of European neutron and muon facilities have agreed on a set of guidelines for development practices, infrastructure, and functional and non-functional product properties. These guidelines have been derived from actual practices in software projects from the EU funded consortium 'Science and Innovation with Neutrons in Europe in 2020' (SINE2020), and have been enriched through extensive literature review. Besides guiding the work of the professional software engineers in our computing groups, we hope to influence scientists who are willing to contribute their own data treatment software to our community. Moreover, this work may also provide inspiration to scientific software development beyond the neutron and muon field.	☑	☐	☐	☐	☐	☐	☑	☐
268	Geotechnical engineering educational modules demonstrating measurement and regulation of soil moisture	With the rapidly changing needs of the construction industry and higher education, Conceive-Design-Implement-Operate (CDIO) educational framework is essential in delivering basics and practical aspects to undergraduate students. This study presents an attempt to explore the use of hardware and software applications for geotechnical engineering education. An example for development of hardware (Arduino UNO) and software applications (Mobile applications) for estimating soil moisture and its regulation is presented. The study provides a new learning environment, where students from civil engineering (major) can work hand to hand with that from electronics (minor) to develop applications for soil moisture measurement. A collaborative workshop is proposed and executed. The workshop includes basic concepts related to hardware (Time domain reflectometry [TDR] probe, Arduino UNO, and Water flow measurement) and software (LabVIEW and Android studio programming platforms) followed by project implementation in a group of 5-6 students. Satisfaction and comprehension were found to increase significantly for subjects (electronics and optics) to students of civil engineering background. The adopted tools represent economical means for application in conventional classrooms as well as laboratories for undergraduate students with a civil engineering background. This integration of the digital and physical world is an attempt at approaching soil mechanics mathematics from technology and arts from a scientific perspective. The examples presented in this study can be useful for developing open-source laboratories within the framework of the undergraduate curriculum in civil engineering.	☑	☐	☐	☐	☐	☐	☑	☐
269	General Variable Neighborhood Search for the optimization of software quality	In the area of Search -Based Software Engineering, software engineering issues are formulated and tackled as optimization problems. Among the problems within this area, the Software Module Clustering Problem (SMCP) consists of finding an organization of a software project that minimizes coupling and maximizes cohesion. Since modular code is easier to understand, the objective of this problem is to increase the quality of software projects, thus increasing their maintainability and reducing the associated costs. In this work we study a recently proposed objective function named Function of Complexity Balance (FCB). Since this problem has been demonstrated to be NP -hard, we propose a new heuristic algorithm based on the General Variable Neighborhood Search (GVNS) schema to tackle the problem. For the GVNS, we propose six different neighborhood structures and categorize them into three different groups. Then, we analyze their contribution to the results obtained by the algorithm. In order to improve the efficiency of the proposed approach, we leverage domain -specific information to perform incremental evaluations of the objective function and to explore only areas of interest in the search space. The proposed algorithm has been tested over a set of real world software repositories, achieving better results than the previous state-of-the-art method, a Hybrid Genetic Algorithm, in terms of both quality and computing times. Furthermore, the relevance of the improvement produced by our proposal has been corroborated by non -parametric statistical tests.	☑	☑	☐	☐	☐	☐	☐	☐
270	FRUIT: Fuzzy Representation of Unbounded and Imprecise Terms in Non-Functional Requirements	It is challenging to verify Non-Functional Requirements (NFRs) because the traditional verification process typically uses the binary concept of true or false to determine whether a requirement has been fulfilled, leaving no space for uncertainty, even in the slightest, such as partially positive or partially negative evaluations. Therefore, it is necessary to consider fuzzy logic to provide the opportunity to verify NFRs in a more flexible manner. To do this, the present manuscript exposes a methodology called the Fuzzy Representation of Unbounded and Imprecise Terms (FRUIT). It expands the PROSE structure, which we first presented in our earlier work, to enable the defining of user NFRs in natural language (NL) when considering fuzzy logic to make more flexible the evaluation of quantitative NFRs and enable the evaluation of qualitative NFRs. To define NFRs under the FRUIT methodology, we need to select the system property, metrics (linguistic variables), and fuzzy set that enable the evaluation of the NFR. A table containing attributes, metrics, and linguistic values is supplied to aid in its definition; the stakeholders and elicitors are tasked with defining the boundaries of each linguistic value. To evaluate the FRUIT methodology, 44 group projects from the required engineering courses offered between 2022 and 2024 at the University of Guayaquil were looked at by experts. According to the results, students decreased the proportion of bad NFRs from approximately 80% to 10%. In conclusion, the results show how important the FRUIT methodology is for helping students to define NFRs in NL while making more flexible the evaluation of NFRs.	☑	☑	☐	☐	☐	☐	☐	☐
271	Fostering Engineering Students? Competences Development Through Lexical Aspect Acquisition Model	The study aims to investigate the effect of enhanced lexical aspect acquisition model on engineering students' competences development, namely such 21st century skills as communication (written or oral), self-organization, systematic and critical thinking, project development and implementation, team work, application of fundamental knowledge, work with information. The researchers have developed a novel approach to fostering lexical aspect acqui-sition through multistage model. The participants of the research were first-year engineering students of the mineral specializations in Saint-Petersburg Mining University. Suggested model-based learning process and conventional-based teaching were used in experimental and control groups accordingly during the spring term 2022. The SPSS software was used to interpret the obtained data. The statistical analysis was carried out with the output of Cronbach's Alpha reliabil-ity test and Pearson correlation test with identification of p-value. The findings showed a significant difference between the level of lexical acquisition in the control and experimental groups proving reliability and validity of the current research.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
272	"Feedback Practices for the Production of the ""Project Report"" Training Genre in Engineering and Science: Exploring the University Classroom"	One of the difficulties generated in the context of confinement due to the SARS-CoV-2 pandemic was the feedback process for written production at the graduate level. This has an essential role in learning, not only regarding the perception of feedback, i.e., giving feedback once the student's work is concluded; but also in the feedforward process that involves accompanying the development of writing by detecting and correcting mistakes (Canabal and Margalef, 2017; Navarro, 2020). As Ahshan (2021) points out, the pandemic forced active teaching and learning strategies to become remote, both synchronous and asynchronous, mediated by educational technologies and platforms for learning management. During the pandemic, the learning of the academic writing process was affected as well. This is a process that already involves different challenges to mediate its elaboration (Carlino, 2013), whether with undergraduate or graduate students (Sabaj et al., 2011). Regarding this obstacle, literature suggests the application of rhetorical moves (Swales, 1990) and genres pedagogy (Venegas et al., 2015; Venegas et al., 2016) as a contribution that facilitates the development of final thesis works. Based on the aforementioned, the question arises: How did the group of graduate students perceive remote feedback from teachers, regarding writing practices, during the elaboration process of the qualitative research project? Therefore, the aim of the study was to understand the perception of remote feedback on writing practices that graduate students received from teachers during the elaboration of the qualitative research project. In order to provide a methodological response, a qualitative, descriptive-phenomenological research was proposed through a case study. The participants were students from the 2018 and 2019 cohorts who took the qualitative analysis elective course of a doctoral education program from a Chilean public university. The classes included group and individual feedback of the writing practices of the qualitative research project, through synchronous educational technologies such as the Zoom platform and Google meet; and also, asynchronous ones such as Ucampus, Youtube and email. Two focus groups were conducted, whose corpus was analyzed under the inspiration of grounded theory (Strauss and Corbin, 2002) with the assistance of Atlas Ti software. Informed consents were applied as ethical criteria for the research. The findings indicate a positive valuation by the students of both the feedback provided through different virtual resources and the modeling of the writing process of the project from video capsules. Furthermore, the asynchronous interactions allowed the prompt communication of the rubrics, before and after the evaluation process. The thematic relevance stands out, having collected the needs of the participants themselves, and the in-depth application of a critical analysis contributed by the group work of a community of researchers in training, which contrasts with the traditional academic training of postgraduate students.	☑	☐	☐	☐	☐	☐	☑	☐
273	Facilitators and inhibitors of Agile methods adoption: Practitioners view	This study provides empirical evidence to the body of knowledge in Agile methods adoption in small, medium, and large organizations in the global context. This research explores facilitators and inhibitors of Agile methods adoption in software development organizations. A survey was conducted among Agile professionals to gather survey data from 52 software organizations in seven countries across the world. This study found many facilitators of Agile adoption to be significant such as customers' dominant issues, encouragement, project champion, highly competent team, use of tools, etc. Similarly a correlation analysis revealed multiple inhibitors as significant: absence of a full set of right Agile practices, absence of customer presence, absence of tracking mechanisms during Agile progress, and failure to determine the role of the client. The present study identifies that an Agile team with high expertise and competence leads to higher quality in software, customer satisfaction along with return on investment (ROI) while a small Agile team increases ease in handling changing requirements, customer satisfaction, reduced delivery time, and increased ROI. Frequent delivery accelerates better control over work, adds to software quality, customer satisfaction, and in shortening delivery time along with increase ROI. It has also been observed that providing essential features early leads to increase in software quality and customer satisfaction. This study confirms that active customer focus leads to better control over work. Further, absence of customer decreases dealing with changing requirements, and customer satisfaction while absence of progress tracking lowers customer satisfaction.	☑	☐	☐	☐	☐	☐	☑	☐
274	External Projects and Partners: Addressing Challenges and Minimizing Risks from the Outset	Software development projects sourced from external organizations can serve as an excellent platform to help build student competencies because they often provide an environment where students can practice applying their knowledge and skills in an authentic context. However, there are many challenges and risks that can jeopardize the successful execution of such projects. In this report, we discuss some of the lessons learned about the pain points encountered by computing faculty with over a decade of experience running a software engineering studio where teams of undergraduate students work on long-term projects sourced from external partners. Our experience is based on working with a mix of project partners with a major emphasis on non-profit and community organizations and non-technical project partners. We focus on a strategy to carefully screen prospective projects to reveal possible challenges in order to avoid or minimize risks that could impact student learning outcomes.	☑	☑	☐	☐	☐	☐	☐	☐
275	Exploring the Integration of Generative AI Tools in Software Testing Education: A Case Study on ChatGPT and Copilot for Preparatory Testing Artifacts in Postgraduate Learning	Software testing education is important for building qualified testing professionals. To ensure that software testing graduates are ready for real-world challenges, it is necessary to integrate modern tools and technologies into the curriculum. With the emergence of Large Language Models (LLMs), their potential use in software engineering has become a focus, but their application in software testing education remains largely unexplored. This study, conducted in the Capstone Project course of a postgraduate software testing program, was carried out over two semesters with two distinct groups of students. A custom-built Travel Application limited to a web platform was used in the first semester. In the second semester, a new set of students worked with an open-source application, offering a larger-scale, multi-platform experience across web, desktop, and mobile platforms. Students initially created preparatory testing artifacts manually as a group deliverable. Following this, they were assigned an individual assignment to generate the same artifacts using LLM tools such as ChatGPT 3.5 in the first semester and Microsoft Copilot in the second. This process directly compared manually created artifacts and those generated using LLMs, leveraging AI for faster outputs. After completion, they responded to a set of assigned questions. The students' responses were assessed using an integrated methodology, including quantitative and qualitative assessments, sentiment analysis to understand emotions, and a thematic approach to extract deeper insights. The findings revealed that while LLMs can assist and augment manual testing efforts, they cannot entirely replace the need for manual testing. By incorporating innovative technology into the curriculum, this study highlights how Generative AI can support active learning, connect theoretical concepts with practical applications, and align educational practices with industry needs.	☑	☑	☐	☐	☐	☐	☐	☐
276	Exploring the Integration of Blockchain and Distributed DevOps for Secure, Transparent, and Traceable Software Development	Distributed DevOps is a software development methodology that aims to integrate the work of development and operations teams without being bound by geographical constraints. This methodology excels in enhancing collaboration and speeding software development. However, it does suffer from a lack of security, transparency, and traceability, which can result in project delays, a lack of trust between stakeholders, and even project failure. This paper addresses these issues of Distributed DevOps by implementing Blockchain technology. In this paper, we propose a novel framework that leverages blockchain technology to address the challenges faced by Distributed DevOps. Through performance analysis, we demonstrate the effectiveness of our framework in a real-world scenario, highlighting its ability to improve transparency, traceability, and the security of the DevOps pipeline. Our findings underscore the potential of blockchain-empowered solutions in revolutionizing DevOps practices. Furthermore, this research offers a practical framework for organizations seeking to optimize their development processes by integrating blockchain technology.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312							RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
277	Experimental Evaluation of Test-Driven Development With Interns Working on a Real Industrial Project	Context: There is still little evidence on differences between Test-Driven Development and Test-Last Development, especially for real-world projects, so their impact on code/test quality is an ongoing research trend. An empirical comparison is presented, with 19 participants working on an industrial project developed for an energy market software company, implementing real-world requirements for one of the company's customers. Objective: Examine the impact of TDD and TLD on quality of the code and the tests. The aim is to evaluate if there is a significant difference in external code quality and test quality between these techniques. Method: The experiment is based on a randomized within-subjects block design, with participants working for three months on the same requirements using different techniques, changed from week to week, within three different competence blocks: Intermediate, Novice and Mixed. The resulting code was verified for process conformance. The participants developed only business logic and were separated from infrastructural concerns. A separate group of code repositories was used to work without unit tests, to verify that the requirements were not too easy for the participants. Also, it was analysed if there is any difference between the code created by shared efforts of developers with different competences and the code created by participants isolated in the competence blocks. The resulting implementations had LOC order of magnitude of 10k. Results: Statistically significant advantage of TDD in terms of external code quality (1.8 fewer bugs) and test quality (5 percentage points higher) than TLD. Additionally, TDD narrows the gap in code coverage between developers from different competence blocks. At the same time, TDD proved to have a considerable entry barrier and was hard to follow strictly, especially by Novices. Still, no significant difference w.r.t. code coverage has been observed between the Intermediate and the Novice developers - as opposed to TLD, which was easier to follow. Lastly, isolating the Intermediate developers from the Novices had significant impact on the code quality. Conclusion:TDD is a recommended technique for software projects with a long horizon or when it is critical to minimize the number of bugs and achieve high code coverage.	☑	☑	☐	☐	☐	☐	☐	☐
278	Experiencing Education 5.0 for Civil Engineering	Nowadays, the construction industry's digital transformation processes are challenging civil engineering education. In general, students and society underestimate the civil engineering field as an outdated and archaic knowledge area. Despite the defragmentation and heterogeneity of the sector, advances have been made on the path for Construction 4.0 and 5.0 scenarios. From this perspective, the CONSTRUCT-Gequaltec group of the Faculty of Engineering of the University of Porto have been sponsoring research and technological development using hardware and software targeting construction management solutions. This paper presents the work in progress mainly connected to the curricular units regarding project management in the Construction Section of the Civil Engineering Department. It also targets assumptions of some expected results across the research and innovation priorities from ECTP Innovative Built Environment and United Nations Sustainable Development Goals (SDGs). Two fundamental elements of the teaching practices are presented as the Construction Sites of the Future (Laboratory -Based Education initiative) and Digital products catalogue targeting Digital Twins in Construction (Project Based Learning initiative). The learning outcomes address increasing knowledge of innovative solutions, delivering skills to deploy 5.0 actions, and targeting humancentred and sustainable attitudes and values.	☑	☑	☐	☐	☐	☐	☐	☐
279	Experiences and challenges from developing cyber-physical systems in industry-academia collaboration	Cyber-physical systems (CPSs) are increasing in developmental complexity. Several emerging technologies, such as Model-based engineering, DevOps, and Artificial intelligence, are expected to alleviate the associated complexity by introducing more advanced capabilities. The AIDOaRt research project investigates how the aforementioned technologies can assist in developing complex CPSs in various industrial use cases. In this paper, we discuss the experiences of industry and academia collaborating to improve the development of complex CPSs through the experiences in the research project. In particular, the paper presents the results of two working groups that examined the challenges of developing complex CPSs from an industrial and academic perspective when considering the previously mentioned technologies. We present five identified challenge areas from developing complex CPSs and discuss them from the perspective of industry and academia: data, modeling, requirements engineering, continuous software and system engineering, as well as intelligence and automation. Furthermore, we highlight practical experience in collaboration from the project via two explicit use cases and connect them to the challenge areas. Finally, we discuss some lessons learned through the collaborations, which might foster future collaborative efforts.	☑	☐	☐	☐	☐	☐	☐	☑
280	Examining Teamwork: Evaluating Individual Contributions in Collaborative Software Engineering Projects	Collaborative learning and group projects are integral to Software Engineering (SE) education as they help prepare students for professional environments where teamwork and collective problem-solving are crucial. Accurately assessing individual contributions within the group project setting remains a significant challenge. Traditional assessment methods often fail to distinguish between individual efforts and collective achievements, particularly in team-based assignments prevalent in SE courses. This paper explores how several factors potentially affect individual participation in group programming projects. We compare metrics for evaluating individual contributions and examine the effect of group formation strategies on collaborative practices. Our data consist of the collaborative artifacts of 77 teams and peer evaluation responses from 154 students across two semesters of a graduate-level software engineering course, providing insights into several research questions, including the effectiveness of different metrics for estimating individual contributions, the distribution of programming tasks among students, and the impact of group composition on task distribution and peer evaluations.	☑	☑	☐	☐	☐	☐	☐	☐
281	Evaluation and Use of a Student-Centered Syllabus for the Software Process Subject in a Postgraduate Course: A Quasi-Experiment	Background: The software-development process is considered a knowledge area in software engineering that allows the performance of activities for the inception of a functional software. In a globalized software context, where companies and organizations have their data and information controlled by applications, it can be seen that the form of construction, that is, the path for which the software is built, has relevance for users. Problem Analysis: Within this reality, it is possible to observe that over the years, scientific research has identified competence deficits for professionals trained in software engineering and more specifically in the software-development processes. Purpose: This work aims to develop and apply a syllabus for the software-development process, with student-centered learning strategies, as it allows for centrality of the teaching and learning process for the student. Methods: The strategies were selected through a literature review, which informed which ones were adopted to intervene in the traditional teaching process in software. Other methodological procedures adopted to identification of the necessary requirements for the development of the syllabus were (i) a survey with the identification and perception of professionals and professors on the research theme, (ii) equivalence mapping of the main documents related to the theme of research at an educational level, and (iii) the identification of how to develop a syllabus. After gathering the requirements, the syllabus was designed and described in each of its parts based on the specialized literature. The evaluation of the syllabus took place in two stages: (i) an expert panel, where experts in the software-development process and student-centered learning were selected, who contributed points to be adjusted for the use of the syllabus, and (ii) the experiment, which was the use of the syllabus in comparison with the traditional approach (use of theoretical classes, exercise list, and written test), through the control and experimental groups. Results: Data generated by using the syllabus in the experiment, as well as data from the control group, were analyzed using the two-tailed Student-t technique. The results achieved in the work demonstrate that there was a considerable gain in learning compared to the control group. Another result point of the syllabus was that through the project strategies, students had contact with a progression of knowledge related to hypothetical cases that simulated reality, which students reported was very good for their knowledge. Conclusions: The use of student-centered strategies can bring benefits in the learning of software-development processes, as it allows a practical application of real scenarios observed in software-development companies.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
282	Evaluating the Impact of Pair Documentation on Requirements Quality in Agile Software Development	A fundamental deliverable of the requirements engineering process is the software requirements specification (SRS) document. A reliable SRS ensures that the final product aligns with the stakeholder's expectations. Yet poorly documented requirements may lead to significant project risks, which include delays, cost overruns, and product failure. Currently, delivering a precise SRS remains a complex task which sometimes result in miscommunication and incomplete information. Such challenges are particularly seen in agile development where the iterative nature of projects requires continuous refinement of the requirements. Consequently, this empirical research aims to evaluate the impact of pair documentation on the quality and effectiveness of SRS documents within agile development teams. Twenty pairs of documentation writers worked into two groups where one group used the pair documentation, i.e., the experimental group, and the other one used conventional documentation i.e., the control group. Both groups were tasked with producing SRS documents for the same project. The outcomes are evaluated against three key criteria of completeness, ambiguity, and consistency of the documents. Our preliminary analysis indicates that the experimental group demonstrated an improvement in document quality with fewer ambiguities, more comprehensive coverage of requirements, and greater consistency across sections. Our research further suggests that collaborative efforts in pair documentation not only enhance the clarity and precision of SRS documents but also may mitigate risks associated with possible miscommunications. In addition, the findings of this study emphasize the potential of pair documentation as a strategy to deliver higher quality SRS documents in dynamic project environments.	☑	☑	☐	☐	☐	☐	☐	☐
283	Evaluating Design Guidelines for Intuitive, Therefore Sustainable, Virtual Reality Authoring Tools	Virtual reality software might be challenging to utilize for beginners and unskilled professionals who do not have a programming or 3D modeling background. Concurrently, there is a knowledge gap in software project design for intuitive virtual reality authoring tools, which were supposed to be easier to use. These tools are frequently insufficient due to a lack of support and standard operating procedures. This study evaluates the validity of fourteen design guidelines for the development of intuitive virtual reality authoring tools. Adopting the Design Science Research approach, a previous study completed the first steps of the protocol by identifying problems, defining solution objectives, and developing and demonstrating the design guidelines. This work evaluates their application in a experiment. A group of engineering students with no prior experience in creating virtual worlds were tasked with examining the design guidelines while using the NVIDIA Omniverse Enterprise as an exemplary use case by following the tutorials in the platform. Previously, the students answered a Likert-scale questionnaire and a focus group interview, with eighteen questions about how they perceived these guidelines. An average score value for the tool was estimated through the questionnaire answers and a correlation analysis and, using the Pearson Correlation Coefficient (PCC), it was confirmed that most guidelines scores behaved as expected and were ranked according to the use-case functionality. The participants understood the guidelines' definition and could decide if they agreed or disagreed with their presence during the experiment. We evaluated that, in accordance with the Design Science Research, the proposed artifact is useful, i.e., the design guidelines for virtual reality authoring tools perform what they are designed to do and are operationally reliable in accomplishing their goals.	☑	☐	☐	☐	☐	☐	☑	☐
284	Evaluating criteria influencing safety performance in metro construction	The study aims to ascertain the important criteria influencing safety awareness and behaviors during the metro construction phase. Through a comprehensive literature review, thirty-five influencing criteria were identified and analyzed as a safety performance evaluation measure using SPSS software. Then, a survey questionnaire was designed to assess the level of agreement among participants about influencing capability of criteria in the Vietnamese construction environment on a five-point Likert scale. Exploratory factor analysis (EFA) revealed six underlying factors, including Education and training, Legal and political, Management, Cost and resource, Work environment, and Technical design. The findings indicated that the group of criteria related to management exerted the most significant influence on occupational safety outcomes. Subsequently, relationships and correlations among the latent factors from the EFA step were examined based on a linear regression analysis framework. The study contributes to a deeper understanding of the criteria affecting safety performance, simultaneously proposing occupational safety measures to advance occupational health and safety management efficiency during the metro construction process. Additionally, the study motivates construction managers to prioritize the implementation of proactive safety management practices within construction projects to enhance worker well-being, improve overall safety conditions, and significantly mitigate occupational risks for construction tradesmen.	☑	☐	☐	☐	☐	☐	☑	☐
285	Enhancing Learning and Collaboration in a Unit Operations Course: Using AI as a Catalyst to Create Engaging Problem-Based Learning Scenarios	This paper presents an innovative approach to problem-based learning (PBL) designed with the aid of ChatGPT in a Unit Operations course. Students were tasked with designing an industrial dryer for specific technological applications of social and economic relevance in Brazil, answering key learning outcomes of the undergraduate program: tackling open-ended problems, employing diverse data gathering strategies, and developing mathematical and simulation skills. One particular aspect of this PBL activity was the use of commercial process simulation software for designing and simulating the dryer. To foster a collaborative learning environment, students were divided into groups with assigned roles, which were evaluated distinctively. This approach helped enhance engagement and involvement and significantly improved learning outcomes. Over 90% of the students reported increased engagement, better teamwork dynamics, and enhanced learning. A feature of this PBL activity was the integration of generative AI (ChatGPT) in diverse simulation scenarios. ChatGPT provided key data for process simulation such as drying curves and particle size distributions, enriching the learning experience by introducing a range of realistic scenarios. This paper details the methodology, implementation, and positive educational outcomes of this approach, highlighting the potential of AI-assisted PBL in enriching chemical engineering and industrial chemistry education.	☑	☐	☐	☐	☐	☐	☐	☑
286	Enablers and inhibitors in Agile Teams - A Case Study Using Challenge Based Learning for Mobile Application Development	Mobile application development requires fast development due to shorter go-to market initiatives. In this context, understanding agile teams is crucial for the mobile software development industry. Thus, identifying and understanding enablers and inhibitors of agile teams in active learning environments has become a research opportunity among the software engineering research community and it is the main goal of this study. To achieve this study goal, we have performed a one-month case study with 50 individuals as part of a two-year mobile development course that teaches iOS development to undergraduate students in a Challenge Based Learning environment. In the course, students experience an agile environment developing mobile applications, dedicating 20 hours a week for the development of real world projects. The case study was conducted in two stages, both of which were held during the COVID-19 pandemic. On the first stage, we have found 33 positive aspects as well as 33 negative aspects reported by participants. However, on the second stage participants have reported 30 positive aspects and 35 negative aspects. The set of both positive aspects and negative aspects partially repeats from the first stage to the second stage. We concluded that team relationship, communication and engagement positively influence mobile application development in agile environments. At the same time, this study also presented negative influences for mobile application development teams, such as teams working remotely (because of COVID-19), lack of knowledge and bad planning.	☑	☐	☐	☐	☐	☐	☐	☑
287	Emulating a Tech Startup in a University Everything but the Code	This paper describes the instructional techniques used in a sequence of courses in the MS program in Software Management at Carnegie Mellon's Silicon Valley campus. Students form 4-5 person teams around their own project ideas, and work through those ideas for two semesters. The class assignments cover everything from ideation of product ideas through financial models for the envisioned company, including identification of the pain points faced by prospective users, the target market, customer validation of the project ideas, creation of a product roadmap, competitive analysis, estimation of engineering effort, strategies for marketing and sales, and financial models. Many of the students intend to become product managers, and these tasks offer experience with what should be done as part of creating a startup business. This paper focuses on the first course, Product Definition and Validation, of the course sequence.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
288	Emotional Awareness in Software Development: Evaluating Software Project Managers' Perspectives	Developers' emotions significantly influence job satisfaction, motivation, productivity, and work quality throughout the software development lifecycle. Previous research highlighted the impact of human factors in various professional domains, such as decision-making, job performance, and project success. However, systematic monitoring and integration into decision-making processes within software development remain underexplored. This study investigates how software project managers approach emotional management in their projects, focusing on identifying emotional variations and their actions in response. Besides, the managers experimented with an emotional polarity-monitoring dashboard tool designed to display developers' emotional states over time. The findings reveal that emotional variations are currently identified in an ad-hoc manner, underscoring the need for a systematic approach to monitor and manage the impact of developers' emotional fluctuations on the development process. Software project managers recognized that a systematic approach such as the one provided by the tool used in this study would have the potential to boost productivity, team cohesion, decision-making, and institutional well-being, emphasizing the value of understanding developers' psychological aspects to enhance the software development process.	☑	☐	☐	☐	☐	☐	☑	☐
289	Effectiveness of Video-based Training for Face-to-face Communication Skills of Software Engineers: Evidence from a Three-year Study	Objectives. Communication skills are crucial for effective software development teams, but those skills are difficult to teach. The goal of our project is to evaluate the effectiveness of teaching face-to-face communication skills using AVW-Space, a platform for video-based learning that provides personalized nudges to support student's engagement during video watching. Participants. The participants in our study are second-year software engineering students. The study was conducted over three years, with students enrolled in a semester-long project course. Study Method. We performed a quasi-experimental study over three years to teach face-to-face communication using AVW-Space, a video-based learning platform. We present the instance of AVW-Space we developed to teach face-to-face communication. Participants watched and commented on 10 videos and later commented on the recording of their own team meeting. In 2020, the participants (n = 50) did not receive nudges, and we use the data collected that year as control. In 2021 (n = 49) and 2022 (n = 48), nudges were provided adaptively to encourage students to write more and higher-quality comments. Findings. The findings from the study show the effectiveness of nudges. We found significant differences in engagement when nudges were provided. Furthermore, there is a causal effect of nudges on the interaction time, the total number of comments written, and the number of high-quality comments, as well as on learning. Finally, participants exposed to nudges reported higher perceived learning. Conclusions. Our research shows the effect of nudges on student engagement and learning while using the instance of AVW-Space for teaching face-to-face communication skills. Future work will explore other soft skills, as well as providing explanations for the decisions made by AVW-Space.	☑	☑	☐	☐	☐	☐	☐	☐
290	Effectiveness of cooperative, collaborative, and interdisciplinary learning guided by software development in Spanish universities	The European Higher Education Area (EHEA) proposes to enhance active learning and student protagonism in order to improve academic performance. In this sense, different methodologies are emerging to create scenarios for self-regulation of their learning. In this study the cooperative, collaborative and interdisciplinary learning methodologies were compared in Spanish universities. The main objectives were to evaluate their effects in higher education and to explore the relationship between perceived group cooperation and self-perceived ability to work in a group, differences between educational Spanish contexts, educational methodologies and gender. To this end, a quasi-experimental design was carried out. Data analysis included the descriptive metrics, correlations and analysis of variance to evaluate the differences among pedagogical methods, their effects on cooperative learning, teamwork outcomes and gender differences, comprising a total of 229 students in Spain from Psychology, Early Childhood Education, Primary Education and Computer Engineering completed the two questionnaires. Results showed that the highest correlation between perceived cooperative activity and self-perceived ability to work in a group was found among computer science students, especially among women, suggesting that the interdisciplinary learning focused on software tool development may be the most effective methodology to improve teamwork and cooperative learning outcomes. Despite these findings concern only to Spanish universities, limiting the generalizability of results, the interdisciplinary methodology seems promising for improving both teaching quality and teamwork skills. The learning methodologies of interdisciplinary projects may therefore need to be implemented within the framework of cooperative and collaborative methodologies.	☑	☐	☐	☐	☐	☐	☐	☑
291	Effectiveness of cooperative, collaborative, and interdisciplinary learning guided by software development in Spanish universities	The European Higher Education Area (EHEA) proposes to enhance active learning and student protagonism in order to improve academic performance. In this sense, different methodologies are emerging to create scenarios for self-regulation of their learning. In this study the cooperative, collaborative and interdisciplinary learning methodologies were compared in Spanish universities. The main objectives were to evaluate their effects in higher education and to explore the relationship between perceived group cooperation and self-perceived ability to work in a group, differences between educational Spanish contexts, educational methodologies and gender. To this end, a quasi-experimental design was carried out. Data analysis included the descriptive metrics, correlations and analysis of variance to evaluate the differences among pedagogical methods, their effects on cooperative learning, teamwork outcomes and gender differences, comprising a total of 229 students in Spain from Psychology, Early Childhood Education, Primary Education and Computer Engineering completed the two questionnaires. Results showed that the highest correlation between perceived cooperative activity and self-perceived ability to work in a group was found among computer science students, especially among women, suggesting that the interdisciplinary learning focused on software tool development may be the most effective methodology to improve teamwork and cooperative learning outcomes. Despite these findings concern only to Spanish universities, limiting the generalizability of results, the interdisciplinary methodology seems promising for improving both teaching quality and teamwork skills. The learning methodologies of interdisciplinary projects may therefore need to be implemented within the framework of cooperative and collaborative methodologies.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
292	DIGITAL CAMPUS AS A TOOL FOR TEACHING ENGLISH IN THE ERA OF DIGITAL EDUCATION	Introduction. The digital transformation of higher education in Russia has caused the need to revise the approaches to language training of university graduates. Aim. The present research aimed to model a Digital Campus with its services and functions as a tool for language training of students in the era of digital transformation of education. Methodology and research methods. The strategic approach was employed as the leading methodological approach, which in the framework of this study is considered as a potential possibility to implement language education at the university using the "Digital Campus" as an important tool for digital transformation of universities. The research was conducted in three stages, involving 2-3 year students of Togliatti State University (113 participants) and school students of Togliatti (157 participants) from July 2022 to January 2023. Empirical research methods were used: analysis, comparison, generalisation, synthesis, modelling. Stage 1 - collection and analysis of the information presented on the websites of the universities in order to highlight the indicators of the digital development of universities. Stage 2 - a survey of schoolchildren to identify their attitude to university admission; a survey of students to determine their motivation to receive language training through digital tools. Stage 3 - modelling of a Digital Campus, its services and functions for students' language training. Results. The authors defined the Digital Campus as an environment based on information and educational digital resources and technologies, computer equipment, telecommunication technologies and software, organisational and methodological support, connecting participants in the learning process (teachers, applicants and students, graduates, enterprises/businesses) for distant educational activities and business operations in accordance with personalised request. The data available on universities' websites were analysed and grouped into seven indicators. The most represented are the services: additional professional education, the use of interactive forms to work with applicants, online support of student and graduate career development, and the project office. Services for the implementation of personalised educational tracks and the digital footprints accumulation during the learning process were less prominent in the course of the research. The results of the survey were the following: there was a general readiness for "applicant - university" interaction through the online environment; a small percentage of divergence of schoolchildren and student opinion regarding their expectations to learn a foreign language online was observed. The Digital Campus and its functions were modelled. Each of the campus services - a student personal account (applicants/students), a teacher personal account, and an employer personal account (enterprises/businesses) - has its own interface and access to the following functional units: "Recruitment and Project System", "Language Courses", "Course Designer", "Resource Management", "E-Learning System", "Courses Expertise Centre". The environment creates unified information and educational space for student language training. Scientific novelty. The approaches to understanding the digital transformation of higher education were systematised, and the indicators of higher education institutions use of digital solutions in the educational process were analytically presented. The article provides the authors' definition of the Digital Campus. The theoretical significance of the article is in the outline and analytical presentation of existing approaches and trends in the digital transformation of higher education. Practical significance. The Digital Campus is an important tool for the implementation of language training and, unlike the common university website, allows real-time interaction between all participants in the educational process: universities - students - enterprises/businesses. The prospects for the study may be aimed at the development of a Digital Campus for engineering students.	☑	☐	☐	☐	☐	☐	☑	☐
293	DevOps and Microservices in Scientific System development: experience on a multi-year industry research project	In this work, we present the experience of a multi-year industry research project where agile methods, microservices and DevOps were applied. Our goal is to validate the hypothesis that the use of microservices would allow computational scientists to work in the more minimalistic prototype-oriented way they prefer while the software engineering team handle the integration. Hence, scientific multidisciplinary systems gain in a twofold way: (i) Subject Matter Experts (SME) use their preferable tools to develop the specific scientific part of the system; (ii) software engineers provide the high quality software code for the system delivery.	☑	☐	☐	☐	☐	☐	☑	☐
294	Design Sprint: Enhancing STEAM and engineering education through agile prototyping and testing ideas	Creating project-based learning experiences in the classroom where students learn in a team to solve complex problems and to develop creative and critical thinking is a challenge. Design Sprint (DS) is an agile methodology (implemented in 5 days) with the goal of creating innovative design based on user needs (User Experience). The objective of this work was to develop an Engineering Drawing classroom experience linked to the context of the current COVID-19 pandemic with the Design Sprint methodology. The experience had to involve the integration of theory and practice, the application of knowledge, the development of both hard and soft skills, and the empowerment of students to conduct research. 56 first-year students following three STEAM degrees at the University of A Coruna participated in this experience. The activities were designed for both face-to-face and remote learning. Microsoft Teams and Moodie were used for tutoring and for monitoring student progress. The Moodie Workshop tool was used for the evaluation of the prototypes that were developed and the projects were evaluated by video. The students defended their projects through a presentation in lightning talk format (Ignite). Evaluation rubrics were used following a triple approach: co-evaluation, hetem-evaluation and self-evaluation. The 3D design of the projects was developed with Autodesk software. A total of 18 projects were developed. Once the projects were completed, a survey was administered to evaluate the levels of student satisfaction. The survey results were very positive. The Design Sprint projects also showed positive effects on grades. The Design Sprint method has promoted an interactive learning environment. In addition to its simplicity, a further advantage of DS method is that all student dedication is planned. Students were therefore less likely to feel overloaded, all of which helps with better time management. The DS methodology is multipurpose, so it can be applied to various fields and subjects.	☑	☑	☐	☐	☐	☐	☐	☐
295	Deploying Optimized Scientific and Engineering Applications on Exascale Systems	Exascale supercomputers are first-of-their-kind instruments with potentially paradigm-shifting capabilities. However, enabling complex applications at scale and high performance can be difficult, requiring hard work and deep technical understanding. Specific programming models are often needed to achieve the desired performance, and, early on, systems are undergoing debugging and rapid development. This makes it challenging to run on new systems quickly following their deployment. The Application Integration (AppInt) area of the Exascale Computing Project was designed to be the integration point between applications, supporting software, system environments, high-performance computing facilities, and vendors. We describe how the AppInt team addressed these challenges while also promoting the use of portable and sustainable programming models and helping harden systems prior to general availability.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
296	Dealing With Cultural Dispersion: a Novel Theoretical Framework for Software Engineering Research and Practice	Software development is fundamentally a team-driven process; researchers in software engineering have identified various human and social factors that can significantly impact it. Culture emerged as a critical element, and the diversity deriving from cultural differences can be highly impactful both positively and negatively. Despite existing knowledge about how culture influences software development, limitations persist. Most importantly, a unified and comprehensive (grounded) theory of how cultural differences influence and are managed in software development has yet to exist. This lack has two significant consequences: (1) it makes research on culture fragmented, leading to the continual definition of new concepts that do not allow state of the art to advance significantly, and (2) it reduces the ability of the research to be transferred to practitioners since there is no framework designed to be understood and used by them. To address the above-mentioned limitation, this work proposed a theoretical framework of "Dealing With Cultural Dispersion," which focuses on challenges and benefits originating from cultural differences and strategies for dealing with them. Such a framework was developed through a qualitative study using an iterative research approach, including interviews and socio-technical grounded theory for data analysis. The proposed framework was designed to reveal the tangible effects of practitioners' culture in software development, allowing software teams to (1) clearly understand the problem and (2) implement the correct strategy for addressing it. Additionally, researchers can use this framework as a foundation to (deductively) develop a more robust and comprehensive theory in this field.	☑	☑	☐	☐	☐	☐	☐	☐
297	Criteria-based comparison of Systems Engineering and Agile Methods A fundamental comparison of two development methods	This paper compares two popular design methods used in product development, namely Systems Engineering (SE) and agile methods. The comparison is based on various criteria, such as ease of implementation, object of innovation, software-dependency, customer relations, human centricity, organizational structure, values, and innovative capability, which were developed in a previous paper. The content of the comparison is supported by a literature review and a workshop with industry professionals. The findings show that both SE and agile methods have similarities and differences based on the criteria examined, and the choice of methodology depends on the project's specific context and objectives. While this paper provides only a general overview of each criterion and their compatibility, the results can assist product development teams in selecting the most appropriate approach for their needs. However, more research is necessary to enhance the effectiveness and efficiency of both approaches. This paper is a useful resource for practitioners and researchers interested in understanding the similarities and differences between SE and Agile and making informed decisions regarding their use.	☑	☐	☐	☐	☐	☐	☑	☐
298	CRISP-NET: Integration of the CRISP-DM Model with Network Analysis	To carry out data analysis, it is necessary to implement a model that guides the process in an orderly and sequential manner, with the aim of maintaining control over software development and its documentation. One of the most widely used tools in the field of data analysis is the Cross-Industry Standard Process for Data Mining (CRISP-DM), which serves as a reference framework for data mining, allowing the identification of patterns and, based on them, supporting informed decision-making. Another tool used for pattern identification and the study of relationships within systems is network analysis (NA), which makes it possible to explore how different components are interconnected. The integration of these tools can be justified and developed under the principles of Situational Method Engineering (SME), which allows for the adaptation and customization of existing methods according to the specific needs of a problem or context. Through SME, it is possible to determine which components of CRISP-DM need to be adjusted to efficiently incorporate NA, ensuring that this integration aligns with the project's objectives in a structured and effective manner. The proposed methodological process was applied in a real working group, which allowed its functionality to be validated, each phase to be documented, and concrete outputs to be generated, demonstrating its usefulness for the development of analytical projects.	☑	☐	☐	☐	☐	☐	☑	☐
299	Countering Hate Speech through Prosocial Student Projects	We discuss misinformation, conspiracy mongering, and hate speech against vulnerable populations as a threat to social justice, democracy, and world peace. Since the hate in vile statements lies in context rather than on specific words, automated detection is hard. Efforts to automate hate speech detection by traditional methods such as logistic regression, decision trees, random forests and linear SVM as well as by Machine Learning methods are making progress but still fall short. We suggest to counter hate speech by empowering empathy. We advocate engaging students in building prosocial apps for smartphones or tablets to help convey empathy and compassion for those vilified by hate speech. Researchers at ETH and the University of Zurich have found that posting supportive messages for victims of hate speech can counter it more effectively than any other strategy. From our experience teaching ethics and teamwork to upper-level Computer Science and Electrical Engineering students and directing CS capstone projects, we submit that students can be encouraged to develop prosocial, thoughtful apps. Ethical, prosocial software design competitions such as the "Ethics in Action: Art or App Design Challenge" sponsored by the University of Alabama at Birmingham could help.	☑	☐	☐	☐	☐	☐	☐	☑
300	Conversion of Sensitive Data to the Observational Medical Outcomes Partnership Common Data Model: Protocol for the Development and Use of Carrot	Background: The use of data standards is low across the health care system, and converting data to a common data model (CDM) is usually required to undertake international research.Onesuchmodel is the Observational Medical Outcomes Partnership (OMOP) CDM. It has gained substantial traction across researchers and those who have developed data platforms. The Observational Health Care Data Sciences and Informatics (OHDSI) partnership manages OMOP and provides many open-source tools to assist in converting data to the OMOP CDM. The challenge, however, is in the skills, knowledge, know-how, and capacity within teams to convert their data to OMOP. The European Health Care Data Evidence Network provided funds to allow data owners to bring in external resources to do the required conversions. The Carrot software (University of Nottingham) is a new set of open-source tools designed to help address these challenges while not requiring data access by external resources. Objective: The use of data protection rules is increasing, and privacy by design is a core principle under the European and UK legislations related to data protection. Our aims for the Carrot software were to have a standardized mechanism for managing the data curation process, capturing the rules used to convert the data, and creating a platform that can reuse rules across projects to drive standardization of process and improve the speed without compromising on quality. Most importantly, we aimed to deliver this design-by-privacy approach without requiring data access to those creating the rules. Methods: The software was developed using Agile approaches by both software engineers and data engineers, who would ultimately use the system. Experts in OMOP were used to ensure the approaches were correct. An incremental release program was initiated to ensure we delivered continuous progress. Results: Carrot has been delivered and used on a project called COVID-Curated and Open Analysis and Research Platform (CO-CONNECT) to assist in the process of allowing datasets to be discovered via a federated platform. It has been used to create over 45,000 rules, and over 5 million patient records have been converted. This has been achieved while maintaining our principle of not allowing access to the underlying data by the team creating the rules. It has also facilitated the reuse of existing rules, with most rules being reused rather than manually curated. Conclusions:Carrot has demonstrated how it can be used alongside existing OHDSI tools with a focus on the mapping stage. The COVID-Curated and Open Analysis and Research Platform project successfully managed to reuse rules across datasets. The approach is valid and brings the benefits expected, with future work continuing to optimize the generation of rules. International Registered Report Identifier (IRRID): RR1-10.2196/60917	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
301	Collaborative Online International Learning (COIL) in chemical engineering: Preparing students for multicultural and international work environments	In today's interconnected society, chemical engineering students must be prepared to work in international and multicultural environments. However, in our experience, current chemical engineering curricula often fail to develop these competencies. This study aims to demonstrate the benefits of Collaborative Online International Learning (COIL) in chemical engineering education. For the first time, the COIL approach has been implemented in a simulation course. In addition to preparing students for international and multicultural work environments, this experience enhances their problem-solving and critical-thinking skills. Unlike other COIL applications, this project allows for multiple valid solutions, though not all are necessarily optimal. After two successful COIL projects involving chemical engineering students from the Universidad de Valladolid (Spain), the Universidad Nacional de Colombia, and the Universidad Auto<acute accent>noma de Nuevo Leo<acute accent>n (Mexico), students reported feeling more confident in their knowledge and abilities, better prepared for multicultural and international work environments, and more capable of performing well in their first job. In both project editions, survey responses to related questions averaged above 4 out of 5. Key takeaways from this work are that, to accomplish the objectives of a COIL, it is essential to define the project timeline in advance, ensure a similar level of knowledge among students, confirm software access, establish a unified communication platform, and conduct individual kickoff meetings for each team. Additionally, effective international collaboration is more likely when no more than 50 % of a team's members come from the same institution.	☑	☑	☐	☐	☐	☐	☐	☐
302	Co-aligning User-Centered Design and Software Engineering Courses: A Case Study	Introducing students to different perspectives and roles in the development process allows them to engage in the work of cross-disciplinary diverse teams and even can enable them to change roles in designer-developer interactions. Industry work often places recent graduates in preexisting polarized relationship dynamics between different participants in the design and development process. This paper describes a two-stage attempt at co-alignment of software engineering and user-centered design courses: from full alignment with topic intersections and joint project to partial alignment through separate activities. We discuss challenges of both ways including time or technical constraints, increased effort from the program developers and instructors, students' and instructors' frustrations. We finalize by describing benefits of providing students with early experience identifying trade-offs between design requirements and architecture and opportunities for diverse group with different background in computer science.	☑	☐	☐	☐	☐	☐	☐	☑
303	Classification of Task Types in Software Development Projects	Managing software development processes is still a serious challenge and offers the possibility of introducing improvements that will reduce the resources needed to successfully complete projects. The article presents the original concept of classification of types of project tasks, which will allow for more beneficial use of the collected data in management support systems in the IT industry. The currently used agile management methods-described in the article-and the fact that changes during the course of projects are inevitable, were the inspiration for creating sets of tasks that occur in software development. Thanks to statistics for generating tasks and aggregating results in an iterative and incremental way, the analysis is more accurate and allows planning the further course of work in the project, selecting the optimal number of employees in task teams, and identifying bottlenecks that may decide on faster completion of the project with success. The use of data from actual software projects in the IT industry made it possible to classify the types of tasks and the necessary values for further work planning, depending on the nature of the planned software development project.	☑	☐	☐	☐	☐	☐	☑	☐
304	Characterizing Team Cognition Within Software Engineering Teams in an Undergraduate Course	Contribution: The study characterizes aspects of cognitive and metacognitive dimensions of team cognition of software development teams in educational settings.Background: The software development industry requires software engineers and developers to work in teams; for this, there is substantial research on teamwork in the context of the organization. However, little is known about it in the context of educational settings, where there is scant research on teamwork in engineering and computing projects.Research Questions: How do students enact teamwork cognitive engagement and metacognitive regulation in the context of a systems analysis and design course?Methodology: The participants were 127 undergraduate students in a systems development course organized into 26 teams. Qualitative categories were derived from a content analysis based on recorded teamwork sessions, which were then quantized and visualized for pattern identification.Findings: Results provide each team's overall cognitive engagement coefficient and metacognitive regulation coefficient. The findings also describe three clusters, each with a description of a selected team to provide further insights into the identified patterns.	☑	☑	☐	☐	☐	☐	☐	☐
305	Challenges and practices of deep learning model reengineering: A case study on computer vision	ContextMany engineering organizations are reimplementing and extending deep neural networks from the research community. We describe this process as deep learning model reengineering. Deep learning model reengineering - reusing, replicating, adapting, and enhancing state-of-the-art deep learning approaches - is challenging for reasons including under-documented reference models, changing requirements, and the cost of implementation and testing.ObjectivePrior work has characterized the challenges of deep learning model development, but as yet we know little about the deep learning model reengineering process and its common challenges. Prior work has examined DL systems from a "product" view, examining defects from projects regardless of the engineers' purpose. Our study is focused on reengineering activities from a "process" view, and focuses on engineers specifically engaged in the reengineering process.MethodOur goal is to understand the characteristics and challenges of deep learning model reengineering. We conducted a mixed-methods case study of this phenomenon, focusing on the context of computer vision. Our results draw from two data sources: defects reported in open-source reeengineering projects, and interviews conducted with practitioners and the leaders of a reengineering team. From the defect data source, we analyzed 348 defects from 27 open-source deep learning projects. Meanwhile, our reengineering team replicated 7 deep learning models over two years; we interviewed 2 open-source contributors, 4 practitioners, and 6 reengineering team leaders to understand their experiences.ResultsOur results describe how deep learning-based computer vision techniques are reengineered, quantitatively analyze the distribution of defects in this process, and qualitatively discuss challenges and practices. We found that most defects (58%) are reported by re-users, and that reproducibility-related defects tend to be discovered during training (68% of them are). Our analysis shows that most environment defects (88%) are interface defects, and most environment defects (46%) are caused by API defects. We found that training defects have diverse symptoms and root causes. We identified four main challenges in the DL reengineering process: model operationalization, performance debugging, portability of DL operations, and customized data pipeline. Integrating our quantitative and qualitative data, we propose a novel reengineering workflow.ConclusionsOur findings inform several conclusion, including: standardizing model reengineering practices, developing validation tools to support model reengineering, automated support beyond manual model reengineering, and measuring additional unknown aspects of model reengineering.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26
IDENTIFICAÇÃO			EXC	MOTIVO					
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
306	C2M: a maturity model for the evaluation of communication in distributed software development	Communication is essential in any software development project, particularly those globally distributed where geographical, temporal, and cultural distance may hinder the effectiveness of communication. The challenges imposed by distance often characterize communication as still one of the main drawbacks of globally distributed projects. Therefore, establishing communication processes and practices is relevant to support a team's work. These processes and practices need to be updated and aligned with the team's needs. Thus, assessing and evaluating the maturity of such communication processes and practices is paramount. This article presents a Communication Maturity Model called C2M which aims to help organizations identify the maturity of communication-related aspects by providing an approach for revealing what practices need to be improved. The model is composed of 4 levels of maturity (causal, partially managed, managed and reflective) and 4 areas of maturity (people, project, organizational and engineering) which are organized into 15 maturity factors, each factor comprising a set of practices. The model has 58 practices and each has its specific objectives. The model was empirically developed and evaluated in three well-defined phases. In the conception phase, methodological procedures (Tertiary Study, Systematic Literature Review, and Interviews) were carried out in order to gather relevant information for designing the first version of the C2M model (alpha version). Then, in the refinement phase, two focus group meetings were held in two organizations in order to identify how effectively the model attends its purpose. The results led to a second version of the C2M model (beta version), analyzed by a survey with experts who assessed the representation of the third version of the C2M model-omega version (evaluation phase). All results achieved so far suggest that the model can assist in discovering the maturity level of the communication processes and practices in globally distributed projects. Future works will focus on developing a software tool to help with self-assessment.	☑	☐	☐	☐	☐	☐	☑	☐
307	Bus Factor In Practice	Bus factor is a metric that identifies how resilient is the project to the sudden engineer turnover. It states the minimal number of engineers that have to be hit by a bus for a project to be stalled. Even though the metric is often discussed in the community, few studies consider its general relevance. Moreover, the existing tools for bus factor estimation focus solely on the data from version control systems, even though there exists other channels for knowledge generation and distribution. With a survey of 269 engineers, we find that the bus factor is perceived as an important problem in collective development, and determine the highest impact channels of knowledge generation and distribution in software development teams. We also propose a multimodal bus factor estimation algorithm that uses data on code reviews and meetings together with the VCS data. We test the algorithm on 13 projects developed at JetBrains and compared its results to the results of the state-of-the-art tool by Avelino et al. against the ground truth collected in a survey of the engineers working on these projects. Our algorithm is slightly better in terms of both predicting the bus factor as well as key developers compared to the results of Avelino et al. Finally, we use the interviews and the surveys to derive a set of best practices to address the bus factor issue and proposals for the possible bus factor assessment tool.	☑	☑	☐	☐	☐	☐	☐	☐
308	BUILDING THE PROFESSIONAL COMPETENCE OF FUTURE PROGRAMMERS USING METHODS AND TOOLS OF FLEXIBLE DEVELOPMENT OF SOFTWARE APPLICATIONS	To support their professional expertise, a modern programmer must constantly follow new technologies, learn new methods of solving problems (best practices), exchange experience, use auxiliary tools that accelerate the development process, should be able to work in a team and develop their knowledge and skills. The task of modern IT education is to meet the demands of the information technology market. Specialists should be provided with proper training, which will give them relevant professional competences. The present paper analyzes modern methodologies of flexible software development and tools, defines professional competences related to software development based on the standard of higher education for the specialists of the Bachelor's academic level in Software engineering specialty. The novelty of the research lies in the justification of the competency-based approach to the training of future programmers This approach involves the use of methods and tools for flexible development of software applications in three stages of project tasks of different types and complexity, which are formed in accordance with certain professional competencies. At Phase 1, students studied flexible methodologies and tools for developing software applications in the Software Design academic discipline. In Phase 2, flexible methodologies and software development tools were used during academic and technological practical training, in particular, students performed a group project according to the Scrum methodology, using Kanban approaches. In Phase 3, students worked individually on the Bachelor's thesis under the guidance of a teacher. The article describes the organization of the work process on the principles of flexible development and flexible learning, presents the results of experimental research, which showed an increase in the level of professional competencies in software development. A statistical analysis of the results of the experiment has been carried out and their significance has been proved.	☑	☐	☐	☐	☐	☐	☐	☑
309	Bridging the Theory-Practice Gap in a Maintenance Programming Course: An Experience Report	This paper presents our experience in teaching a maintenance programming course with the aim of bridging the gap between theory and practice, a recurring issue in previous course offerings. To achieve this goal, we implemented active learning strategies within an active learning classroom setting and redesigned the project work. Our approach involves peer learning and teamwork activities to cover various aspects of legacy code maintenance. For the project work, we adopted an open-ended approach that allowed students to choose their legacy code projects, which could be open source software or a previous software project they had worked on. Analysis of students' feedback and project reports highlighted the effectiveness of our approach in bridging the gap between theory and practice. We believe that our approach had the potential to enhance students' engagement and critical thinking abilities, as well as improve practical maintenance skills relevant to their future careers.	☑	☑	☐	☐	☐	☐	☐	☐
310	Branch Drift: A Visually Explainable Metric for Consistency Monitoring in Collaborative Software Development	Modern software engineering uses branch-based workflows for collaborative development on a shared codebase. A branch may represent a task or feature within the responsibilities of a specific developer or team. When branches are merged, unanticipated conflicts occur frequently. Their resolution is time-consuming and costly. Depending on the development process, a branch exists for a few hours or up to several months. With increasing time and many branches, keeping track of potential merge conflicts becomes difficult. Typically, developers are unaware of these conflicts until the branches are merged. Issues solvable through joint reviews, discussion, and collaboration are therefore identified and resolved late. It is evident that the effort required to resolve merge conflicts poses an organizational technical debt. Project managers and team leaders must be able to track, analyze, and interpret the merge conflict potential to call for timely actions. Therefore, a metric is required. This work transfers the drift metric for inconsistency analysis from model-driven software engineering to branch-based software development. The resulting branch-drift metric summarizes the merge conflict potential of a development project in a numeric value. Its calculation is configurable to fit any project's requirements. The branch-drift is based on an intuitive and explainable geometric representation, easing the analysis of conflict patterns and outliers. We evaluate the branch-drift by quantitatively analyzing and discussing the drift values, trends, and patterns for 21 large-scale open-source repositories over three months. The results of this work encourage and enable the continuous and configurable branch-drift calculation for arbitrary Git repositories. The developed tools and gathered data are publicly available.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
311	Blockchain-Based Software Effort Estimation: An Empirical Study	Context: The success or failure of any software development project significantly depends on the accuracy of its effort estimates. Software development effort estimation is the foundation for project bidding, budgeting, planning, and cost control. Problem: The literature shows that a lot of work has been done on software effort estimation. But still, there is a need for improvement in effort estimation by introducing new methodologies. The structured group-based and analogy-based effort estimations are the widely used estimation methods. Nevertheless, there are several shortcomings of using these methods such as lack of experts, lack of historical data, and biasness in expert opinion, which negatively affect the estimation results. Motivation: With the advancement of technologies, such limitations could be overcome. Such as leveraging the applicability of blockchain in several domains such as improvement in the software development process and network security. Method: In this article, we have proposed a Blockchain-Based Software Effort Estimation (BBSEE) methodology to improve the software effort estimation. We employ the proposed method using Web and blockchain technologies. Moreover, we also proposed evaluation criteria to assess the efficacy of the proposed method in terms of Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE) and percentage of successful predictions falling (PRED (25)). Result: We performed several case studies and analyses of expert opinions of 52 organizations to present the efficacy of the proposed method. Conclusion: We observe that the BBSEE method outperforms expert judgment and analogy-based effort estimation methodologies in terms of software effort estimation.	☑	☑	☐	☐	☐	☐	☐	☐
312	BIM Use to Support the Technical Tasks in Construction Projects: A Case Study	BIM use has proved to be inevitable in construction project phases, starting from the project initiation phase and ending with the project closeout phase. Through the use of 4D/5D/6D models in processes of the construction project such as scheduling, bills of quantities (BoQs), and environmental analysis, BIM provides massive assistance to all project participants. In addition, BIM has been utilized in facility management through the use of 7D models. These well-established benefits positively impact construction project variables: cost, time, and quality. Despite these invaluable contributions in construction projects, BIM is not extensively applied in the Gulf Cooperation Council (GCC) of Arab countries, especially in small/medium-scale construction projects. BIM use at these project scales is mainly exclusive to the design phases. The present use of BIM in GCC construction projects leads to the missing of numerous opportunities, which negatively affects project delivery methods, project timelines, and, consequently, project budgets. To shed more light on this subject, this research paper attempts to answer emerging questions, such as the following: How do BIM and other software programs support the technical team of construction projects? Do these software programs have a direct impact on construction project management, in terms of project timelines and work changes? Does BIM add unnecessary cost to the project budget? This research paper proposes four technical task groups used in construction project phases to evaluate the use of BIM and the associated software programs, namely technical work, Time and Accuracy, budget, and BoQ. These four task groups, moreover, have been ratified and are the subject of consensus by project managers and senior engineers in GCC construction projects, who provided the research primary data through interviews and field surveys. Qualitative and quantitative analyses were conducted to investigate and analyze the present status of BIM use in construction projects, and subsequently, to establish a basis to propose improvements to support the technical team of construction projects. Two approaches are proposed to overcome the obstacles that hinder BIM adoption in GCC countries. The first one is a top-down approach, in which BIM use and BIM documents are mandatory in construction projects. The second approach is a bottom-up approach, where having more BIM professionals in the GCC market will facilitate the transformation to BIM.	☑	☐	☐	☐	☐	☐	☑	☐
313	Assessing Delay Factors and Mitigation Measures in Building Maintenance Works	When it comes to the operation of a construction industry, building maintenance is critical. Any structure must be well-maintained to prevent minor problems from becoming major ones. Delayed building maintenance may result in increased repair costs as well as a reduction in building lifespan. To avoid this problem, research has been conducted to identify the factors that cause building maintenance delays and mitigation measures to resolve building maintenance delays. To achieve the study's objectives, extensive literature research and a questionnaire survey were conducted. Based on previous literature research, accumulated data was used to create a questionnaire for 250 Malaysian building maintenance delay factors and mitigation measures. A group of professionals and experts, including top management, project managers, engineers, architects, assistant engineers, technicians, human resources and administration, and others, participated in this survey. SPSS statistical software was used to analyse the collected data using the average mean index method and correlation. According to the data, the three highest mean index values from the factors that cause maintenance work delays are an increase in maintenance costs as the building age increases, an incorrect timing for work can incur higher maintenance costs when a critical job item is overlooked, and improper planning and budgeting. For correlation analysis, it was found that the correlations amongst the five groups of delay factors are having strong relationship with Pearson correlation value of 0.5. While for mitigation measures of the delay, it was found that three highest ranks are prefer other organizational structure follows by policy and objective and good communication. The findings can assist construction practitioners in prioritising their maintenance works.	☑	☐	☐	☐	☐	☐	☑	☐
314	Architecting a decision support system for continuing supervision of commercial in-space servicing	The rapid development of in-space servicing technology and other novel space capabilities requires robust and transparent governance frameworks to ensure long-term space sustainability and adherence to international regulations, notably Article VI of the Outer Space Treaty. Article VI requires that signatory states provide continuing supervision over non-governmental space activities, a mandate becoming increasingly more challenging to fulfill due to the accelerating pace of commercial space innovations. In previous work published by the authors, a Systems Architecture Framework analysis investigated the governance of in-space servicing in the U.S. and the corresponding Stakeholder Need misalignments with current authorization and supervision processes. The initial research provided insights into the apparent Need for a Decision Support System addressing the practical challenges faced in the operational supervision of in-space servicing activities. In response, this paper roadmaps the application of the Environment-Vulnerability-Decision-Technology (EVDT) systems engineering framework into the realm of space sustainability challenges, such as for authorization and supervision of commercial in-space servicing. Originally conceived by the Space Enabled research group at MIT's Media Lab, the EVDT framework has demonstrated its effectiveness in facilitating sustainable development decision-making through analysis of complex socio-environmental-technical systems across various terrestrial applications. Historical uses of EVDT span across aiding flood resilience in Indonesia, promoting mangrove preservation in Brazil, managing invasive plant species in Benin, revitalizing cranberry wetlands in the U.S., analyzing environmental injustice in prison landscapes, and urban planning strategies during the pandemic. Most recently, the inaugural adaptation of the EVDT framework to the space domain shows potential to enhance collision avoidance operation decisions for a Stakeholder within NASA. This paper proposes the expansion of the EVDT framework to broader space sustainability challenges, focusing on continuing supervision as the primary use-case, where this prototype's capability to model and analyze hypothetical commercial in-space satellite servicing missions under U.S. jurisdiction will demonstrate the potential of EVDT to enhance space situational awareness (SSA) and space domain awareness. These operations are critical for collision prediction and consequence, risk assessment, and the implementation of sustainable operational practices. We introduce the plan for developing the Continuing Supervision EVDT software prototype, using a MATLAB-based method characterized by a modular architecture to facilitate integration and extension of functionality. The paper also introduces terminology, key concepts, objectives, and the use of the Systems Architecture Framework method within the EVDT software development process. The software design enables Stakeholders to custom-build and adapt their models to different space sustainability scenarios, improving code reuse, reducing development time, and simplifying interactions for external users and future space-based EVDT projects. The implementation of this Decision Support System has the potential to influence the authorization and supervision of novel space missions and the evolution of supporting SSA technologies, ultimately contributing to the responsible and sustainable use of the space environment.	☑	☑	☐	☐	☐	☐	☐	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
315	ArchHypo: Managing Software Architecture Uncertainty Using Hypotheses Engineering	Uncertainty is present in software architecture decisions due to a lack of knowledge about the requirements and the solutions involved. However, this uncertainty is usually not made explicit, and decisions can be made based on unproven premises or false assumptions. This paper focuses on a technique called ArchHypo that uses hypotheses engineering to manage uncertainties related to software architecture. It proposes formulating a technical plan based on each hypothesis' assessment, incorporating measures able to mitigate its impact and reduce uncertainty. To evaluate the proposed technique, this paper reports an application of the technique in a mission-critical project that faced several technical challenges. Conclusions were based on data extracted from the project documentation and a questionnaire answered by all team members. As a result, the application of ArchHypo provided a structured approach to dividing the architectural work through iterations, which facilitated architectural decision-making. However, further research is needed to fully understand its impact across different contexts. On the other hand, the team identified the learning curve and process adjustments required for ArchHypo's adoption as significant challenges that could hinder its widespread adoption. In conclusion, the evidence found in this study indicates that the technique has the potential to provide a suitable way to manage the uncertainties related to software architecture, facilitating the strategic postponement of decisions while addressing their potential impact.	☑	☑	☐	☐	☐	☐	☐	☐
316	APPLICATION OF DIGITAL TECHNOLOGIES FOR VIRTUAL RECONSTRUCTION OF THE ROMAN VILLA OF SALAR (GRANADA, SPAIN): AN EXAMPLE OF ARCHAEOLOGICAL HERITAGE TRANSFER	The mechanical earthworks caused by the construction of a wastewater treatment plant in the municipality of Salar (Granada, Spain) (Fig. 1) in November 2004, brought to light various material remains dating from Roman times. After several years of archaeological interventions, and fundamentally, since the archaeological direction was taken over by a scientific team from the University of Granada in 2016, the Roman villa of Salar has been placed among one of the most important agricultural facilities of Roman Hispania, thanks to its exceptional architecture and the good conservation of its decorative elements as well. Only a large part of the aristocratic residence has been excavated. This constitutes a minimal section of what must have been an enormous rural exploitation, in view of the wealth and luxury shown by the excavated sections of the pars urbana corresponding to the Late Empire. The villa's construction during the 4th century AD entailed a profound transformation of the landscape of the Canuto site. The articulation of the different rooms of the villa on terraces, on at least three different levels, made it necessary to carry out important work to lower and cut the terrain, which had a decisive effect on the buildings described above. The architecture and decorative elements of the pars urbana, constructed according to the orders of its dominus, help us to understand the social realities of this rural enclave in the 4th century AD and the magnitude of the changes which took place (Fig. 2). Next to the peristyle is an apsidal space that opens directly onto the northwestern ambulacrum in the central part, on the same axis as the entrance to the triclinium. Of these rooms, the most outstanding is located on an axis with the peristyle and the triclinium, marking the spaces' profound symmetry and theatricality. Regarding the research project, since its inception, it has been characterised by pursuing three fundamental objectives: research, conservation and transfer of the knowledge generated to society, based on an interdisciplinary work methodology, where the implementation of digital techniques for archaeological documentation has been an essential basis for its development (Rosa Henestrosa, Rom & aacute;n Punz & oacute;n, Moreno-Alcaide & Ruiz Montes, 2022). The methodology used in the reconstruction is based on the criteria established for Virtual Archaeology, based on the Seville Principles (L & oacute;pez-Menchero & Grande, 2011). To do this, we formed an interdisciplinary team made up of professionals from archaeology, architecture and engineering, which has made it possible to generate a model with high historical rigour and great technical quality. Starting from the previously analysed work, generated from the interpretations of the archaeological excavations, a new planimetry was developed in vector Computer-Aided Design (CAD) format that would later allow it to be used as a basis for the three-dimensional (3D) modelling of the Roman villa (Fig. 5). With this vector planimetry, the hypothetical architectural sections of the building were also created, as well as the geometric bases for the geometric mosaics that the villa had. The first volumetric fit in the 3D modelling phase also served to answer questions related to the heights of the rooms and the resolution of the building's roofs. The Autodesk 3D Studio Max software was used to create the 3D modelling, and the Adobe Photoshop and Adobe Substance 3D Painter were used to create the textures.	☑	☐	☐	☐	☐	☐	☑	☐
317	An Empirical Study on Usage and Perceptions of LLMs in a Software Engineering Project	Large Language Models (LLMs) represent a leap in artificial intelligence, excelling in tasks using human language(s). Although the main focus of general-purpose LLMs is not code generation, they have shown promising results in the domain. However, the usefulness of LLMs in an academic software engineering project has not been fully explored yet. In this study, we explore the usefulness of LLMs for 214 students working in teams consisting of up to six members. Notably, in the academic course through which this study is conducted, students were encouraged to integrate LLMs into their development tool-chain, in contrast to most other academic courses that explicitly prohibit the use of LLMs. In this paper, we analyze the AI-generated code, prompts used for code generation, and the human intervention levels to integrate the code into the code base. We also conduct a perception study to gain insights into the perceived usefulness, influencing factors, and future outlook of LLM from a computer science student's perspective. Our findings suggest that LLMs can play a crucial role in the early stages of software development, especially in generating foundational code structures, and helping with syntax and error debugging. These insights provide us with a framework on how to effectively utilize LLMs as a tool to enhance the productivity of software engineering students, and highlight the necessity of shifting the educational focus toward preparing students for successful human-AI collaboration.	☑	☑	☐	☐	☐	☐	☐	☐
318	An Empirical Study of the Content and Quality of Sprint Retrospectives in Undergraduate Team Software Projects	The retrospective, or retro, is a fundamental component of the Agile process, widely used in both software engineering courses and industry. In a retro, teams conic together at the end of a sprint to reflect on their team's performance. We conducted an empirical study to explore three research questions concerning retros in undergraduate team projects: (1) What do students reflect on? (2) What is the quality of their reflections? (3) How do teams' retros vary in terms of content and quality? Our study analyzed a corpus of 963 statements documented in the retros of 32 undergraduate software teams (n = 182 students) enrolled in four software engineering courses at two North American universities. A content analysis revealed that teams reflected most often on their work, communication, and collaboration practices. Nearly a third of teams' reflections focused on their general work practices, while nearly half focused on specific areas of the software development lifecycle-most prominently, pull requests, issues, and coding/testing/debugging. An analysis of the quality of teams' retro reflections showed that only 13% provided justification for a strategy to be stopped, continued, or started. An analysis of team-by-team results indicated significant differences in teams' retro content and quality. We compare these results to previous studies of retros in academia and industry; and consider their implications for software engineering education.	☑	☑	☐	☐	☐	☐	☐	☐
319	An Empirical Investigation Into the Influence of Software Communities' Cultural and on	Estimating and understanding software development productivity represent crucial tasks for researchers and practitioners. Although different works focused on evaluating the impact of human factors on productivity, a few explored the influence of cultural/geographical diversity in software development communities. More par-ticularly, all previous treatise addresses cultural aspects as abstract concepts without providing a quantitative representation. Improved knowledge of these matters might help project managers to assemble more productive teams and tool vendors to design software analytics toolkits that may better estimate productivity. This paper has the goal of enlarging the existing body of knowledge on the factors affecting productivity by focusing on cultural and geographical dispersion of a development community-namely, how diverse a community is in terms of cultural attitudes and geographical collocation of the members who belong to it. To reach this goal, we performed a mixed-method empirical study. First, we built a statistical model relating dispersion metrics with the productivity of 25 open-source communities on Orthus. Then, we performed a confirmatory survey with 140 practitioners. The key results of our study indicate that cultural and geographical dispersion considerably impact productivity, thus encouraging managers and practitioners to consider such aspects during all the phases of the software development lifecycle. We conclude our paper by elaborating on the main insights from our analyses and instilling implications that may drive further research.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
320	An Eco-System Approach to Project-Based Learning in Software Engineering Education	Contribution: This article identifies the participation of external stakeholders as a key contributing factor for positive outcomes in project-based software engineering courses. A model for overlapping virtuous circles of lasting positive impact on both stakeholders and students from such courses is proposed. Background: Project-based courses are widespread in software engineering education, and there are numerous designs for such courses presented in literature. It is found that the needs and motivations of external stakeholders, from industry and government sectors, in these courses has received limited attention in related work. Intended Outcomes: A course design that prepares students for graduate level studies and professional life, through close proximity to external stakeholders in a highly realistic setting, working on "live" projects. Application Design: Building on a long tradition of university-industry collaboration dating back to 1977, as well as findings in related work, students are assigned to live projects proposed by external stakeholders from industry and government, working in close proximity with their respective stakeholders throughout the project. The course places great emphasis on coaching over instruction, treating the many unforeseen challenges of such projects as a valuable part of the learning experience. Findings: Based on interviews with stakeholders and students, it is found that stakeholder and student outcomes are interdependent and build upon one another, and that positive outcomes for both groups are necessary for the sustainability of the course over multiple iterations.	☑	☐	☐	☐	☐	☐	☐	☑
321	An Actionable Framework for Understanding and Improving Talent Retention as a Competitive Advantage in IT Organizations	In the rapidly evolving global business landscape, the demand for software has intensified competition among organizations, leading to challenges in retaining highly qualified IT members in software organizations. One of the problems faced by IT organizations is the retention of these strategic professionals, also known as talent. This work presents an actionable framework for Talent Retention (TR) used in IT organizations. It is based on our findings from interviews performed with 21 IT managers. The TR Framework is our main research outcome. Our framework encompasses a set of factors, contextual characteristics, barriers, strategies, and coping mechanisms. Our findings indicated that software engineers can be differentiated from other professional groups, and beyond competitive salaries, other elements for retaining talent in IT organizations should be considered, such as psychological safety, work-life balance, a positive work environment, innovative and challenging projects, and flexible work. A better understanding of factors could guide IT managers in improving talent management processes by addressing Software Engineering challenges, identifying important elements, and exploring strategies at the individual, team, and organizational levels.	☑	☐	☐	☐	☐	☐	☑	☐
322	Aligning the Learning Experience in a Project-Based Course: Lessons Learned from the Redesign of a Programming Lab	In teaching and training the next generation of software engineers, programming labs with students working together in small groups provide the opportunity to obtain hands-on experience for software projects involving multiple developers. However, more than other types of courses, programming labs face some challenges in providing a similar learning outcome for all students. Based on feedback and own experience from various iterations of the programming lab at TU Dortmund University, we identified that the learning experience varies significantly due to heterogeneous prior knowledge, experience levels, and personality traits of both students and tutors. In this experience report, we present our approach towards aligning the learning experience by applying three different didactic improvements based on well-studied concepts: (1) the idea of worked-out examples is transferred to teaching the software development process by providing a small software application with all corresponding artefacts like diagrams, program code and documentation, focusing on their relationships and development activities. (2) Goal-oriented and structured learning is used to define learning outcomes for every group meeting as a common ground, while audience response systems are utilized to motivate the attendance and allow students to self-reflect on their knowledge and competence level. (3) We harmonize the role of tutors by holding dedicated teaching workshops for tutors' responsibilities in the programming lab. The different approaches are evaluated based on surveys for students and tutors over three iterations of the programming lab at TU Dortmund University. Both sides' positive responses and feedback resulted in an enumeration of lessons learned as recommendations and support for other similar courses.	☑	☑	☐	☐	☐	☐	☐	☐
323	Aligning Software Engineering Teaching Strategies and Practices with Industrial Needs	Several approaches have been proposed to reduce the gap between software engineering education and the needs and practices of the software industry. Many of them aim to promote a more active learning attitude in students and provide them with more realistic experiences, thus recreating industry software development environments and collaborative development and, in some cases, with the involvement of companies mainly acting as potential customers. Since many degree courses typically offer separate subjects to teach requirements engineering, analysis and design, coding, or validation, the integration of all these phases normally necessitates experience in a project context and is usually carried out in a final year project. The approach described in this article benefits from the close involvement of a software house company which goes beyond the common involvement of a potential customer. Students are integrated into distributed teams comprising students, teachers and IT professionals. Teams follow the agile Scrum methodology and use the OutSystems low-code development platform providing students with the experience of an almost real scenario. The results show that this approach complements the knowledge and practice acquired in course subjects, develops the students' technical and non-technical skills, such as commitment, teamwork, and communication, and initiates them in the methodologies and development strategies used in these companies. The feedback from the teachers involved, software companies and students was very positive.	☑	☐	☐	☐	☐	☐	☐	☑
324	Airport terminal building capacity evaluation using queuing system	Queues or waiting lines are a natural occurrence in the everyday lives of consumers and the process of every business. Being customers' first point of contact with the business, a customer's experience in the queue becomes a determining factor of their first impression of the business. Queuing provides the cornerstone of efficiency to businesses as they assist employees and managers in tracking, prioritising, and ensuring the delivery of services and transactions. Inefficiencies in queues are undesired as they can result in substantial losses to a business, such as bad reputation and loss of customers due to balking or reneging behaviour.Previous research has shown that the application of queuing theory enables businesses to analyse their queuing system and the trends of demand for their services. This allows the business to effectively identify measures to improve their queuing system and serve demand at the desired level of service.In this paper, where it examined Cairo International Airport (CAI)'s existing departure queue system and benchmarked it against the optimum wait time suggested in the International Air Transport Association's (IATA) Level of Service (LoS) concept. The application of Kendall-Lee's Notation is applied to describe the existing queuing system of the airport. It also suggested areas of improvement after the analysis and recommended that the CAI focus on improving service time for its Check-in, Security, and Boarding process. Although the Immigration process has met IATA's recommended optimal wait time, training could be provided to employees to enable them to progressively work towards a better service time and prepare for the future higher traffic volume.The team has also suggested for CAI to introduce autonomous technology for the departure process and software which analyses passenger flow and projects a forecast value based on the airport growth trends.(c) 2022 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Engineering, Alexandria University This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/ licenses/by-nc-nd/4.0/).	☑	☐	☐	☐	☐	☐	☑	☐
325	Agile Software Engineering Capstone Courses: Exploring the Impact of Gender	The wide adoption of agile methodologies in software development necessitates an educational approach that prepares students for industry practices. This study aims to explore an agile capstone course, focusing on student learning experiences in teamwork while examining the gender dynamics affecting team roles and engagement. We conducted a comprehensive study of the capstone course "Software Engineering with Project Work" at the University of Oslo, analyzing 27 presentations and interviewing 10 students and two teaching assistants. The course is designed to immerse informatics students of different specializations in agile methodologies and collaborative project work. Our findings suggest that the students learn essential collaboration and technical skills. However, during their final presentations, significant gender differences emerged in the topics discussed: women predominantly spoke about design and process, whereas men focused more on technical aspects such as architecture, technical solutions, and algorithms. Our results underscore the need for educational strategies that promote inclusivity and equal participation in agile software development courses.	☑	☐	☐	☐	☐	☐	☐	☑

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO				MOTIVO						
ID	TÍTULO	RESUMO	EXC	DUP	CE1	CE2	CE3	CE4	CE5	CE6
326	Agile Software Development Lifecycle and Containerization Technology for CubeSat Command and Data Handling Module Implementation	As a subclass of nanosatellites, CubeSats have changed the game's rules in the scientific research industry and the development of new space technologies. The main success factors are their cost effectiveness, relative ease of production, and predictable life cycle. CubeSats are very important for training future engineers: bachelor's and master's students of universities. At the same time, using CubeSats is a cost-effective method of nearest space exploration and scientific work. However, many issues are related to efficient time-limited development, software and system-level quality assurance, maintenance, and software reuse. In order to increase the flexibility and reduce the complexity of CubeSat projects, this article proposes a "hybrid" development model that combines the strengths of two approaches: the agile-a-like model for software and the waterfall model for hardware. The paper proposes a new computing platform solution, "Falco SBC/CDHM", based on Microchip (Atmel) ATSAMV71Q21 with improved performance. This type of platform emphasizes low-cost space hardware that can compete with space-grade platforms. The paper substantiates the architecture of onboard software based on microservices and containerization to break down complex software into relatively simple components that undergraduates and graduates can handle within their Master's studies, and postgraduates can use for scientific space projects. The checking of the effectiveness of the microservice architecture and the new proposed platform was carried out experimentally, involving the time spent on executing three typical algorithms of different algorithmic complexities. Algorithms were implemented using native C (Bare-metal) and WASM3 on FreeRTOS containers on two platforms, and performance was measured on both "Falco" and "Pi Pico" by Raspberry. The experiment confirmed the feasibility of the complex application of the "hybrid" development model and microservices and container-based architecture. The proposed approach makes it possible to develop complex software in teams of inexperienced students, minimize risks, provide reusability, and thus increase the attractiveness of CubeSat student projects.	☑	☐	☐	☐	☐	☐	☑	☐
327	Agile software development approach for 'ad-hoc' IT projects	Restrictive Scrum assumptions make the effectiveness of this approach debatable in projects deviating from typical execution conditions. This article delivers a comprehensive software development approach for both academic and commercial Information Technology (IT) projects effectuated by teams that are hampered by significantly unsystematic participation of project members and mercurial internal communication. The nature of 'ad -hoc' projects imposes another level of difficulty in terms of both managing the conduct of such a project and ensuring the quality of the end product. Multicyclic action research enabled a gradual adaptation of the Scrum approach to support such project conditions. This study introduces major alterations to Sprint implementation and minor enhancements within the documentation process to streamline knowledge sharing among Development Team members. Proposed key alterations include the evolution of Daily Scrum towards Weekly Scrum, the possibility of extending Sprints length, the eventuality to switch team members during Sprint due to substantial failure to meet deadlines, having at least two team members responsible for a single Product Backlog Item (PBI) at all times, as well as exclusion of Burndown Chart in favor of Development Team members updating their working time. Positive validation of enhancements in mixed settings confirms that the generic Scrum framework can be adapted to support highly volatile projects. The proposed approach is suitable not only for carrying out software development initiatives that rely heavily on the skills of external experts and/or volunteers. It also supports traditional Scrum teams that seek to reduce their exposure to risk arising from organizational changes.	☑	☐	☐	☐	☐	☐	☑	☐
328	Agile Effort Estimation Usage in the Sri Lankan Software Industry	Accurate effort estimation in the software development industry remains a significant challenge due to requirements' complexities, technology variability, and insufficient skilled members. To provide up-to-date insights on the state of effort estimation practices in Sri Lanka, we surveyed agile practitioners to identify the effort estimation techniques, types of metrics employed, levels of accuracy, and reasons for inaccuracies in estimates. Our analysis of 93 valid responses reveals that Planning Poker was the most popular estimation technique at 50.5%, while story points were the most widely used metric utilized by 61.3% of participants. Expert estimation was employed by 30% of respondents, and man-hours were used by 23.7%. The combination of Planning Poker and story points was most used, with a prevalence of 70.2%. Regarding the accuracy of estimations, respondents who used a combination of Planning Poker (61.7%) and expert estimation (25.5%) could complete their work within the estimated time without any extra effort. The top three categories of inaccurate estimates were quality-related, project management, and team-related issues.	☑	☐	☐	☐	☐	☐	☑	☐
329	Adopting Scrum in Hybrid Settings, in a University Course Project: Reflections and Recommendations	Agile software development (ASD) methods such as Scrum promote collocated team collaboration. However, adopting Scrum in hybrid work environments can pose challenges. Therefore, understanding the positive aspects and challenges of adopting Scrum in hybrid settings and the relevant agile tools and practices is important. Software engineering education should also prepare students with the necessary knowledge and skills for implementing ASD methods in hybrid settings. This study provides insight into implementing Scrum in a hybrid setting, in a course project, discussing its positive aspects, challenges, and relevant project management and communication tools. We used a questionnaire to collect feedback from 12 scrum teams, made up of 59 master's degree students at the University of Oulu, who developed student well-being prototypes in four sprints. We applied thematic analysis to analyze the teams' feedback. We found 16 challenges, 12 positive aspects, and varying ASD practices and tools. Key challenges included scheduling conflicts, communication issues, and estimating efforts. Discord and Microsoft Teams were popular for communication, while JIRA and Trello were favored for agile project management. This study complements software engineering literature with new empirical findings. It reveals practices and tools applicable to software practitioners and start-up teams for effectively adopting scrum in hybrid settings. It also contributes by illustrating the adoption of scrum course projects in hybrid settings and providing recommendations to enhance the design of similar courses. It identifies research gaps, such as the need for tools and practices to improve collaboration and the potential of asynchronous chats in hybrid environments.	☑	☑	☐	☐	☐	☐	☐	☐
330	Acrobats and Safety Nets: Problematizing Large-Scale Agile Software Development	Agile development methods have become a standard in the software industry, including in large-scale projects. These methods share a set of underlying assumptions that distinguish them from more traditional plan-driven approaches. In this article, we adopt Alvesson and Sandberg's problematization approach to challenge three key assumptions that are prevalent in the large-scale agile literature: (1) agile and plan-driven methods are mutually exclusive; (2) self-managing and hierarchically organized teams are mutually exclusive; and (3) agile methods can scale through simple linear composition. Using a longitudinal case study of large-scale agile development, we describe a series of trigger events and episodes whereby the agile approach was tailored to address the needs of the large-scale development context, which was very much at odds with these fundamental assumptions. We develop a set of new underlying assumptions which suggest that agile and plan-driven practices are mutually enabling and necessary for coordination and scaling in large-scale agile projects. We develop nine propositions for large-scale agile projects based on these new alternative underlying assumptions. Finally, we summarize our theoretical contribution in a generic process model of continuously adjusting agile and plan-driven practices in order to accommodate process challenges in large-scale agile projects.	☑	☐	☐	☐	☐	☐	☑	☐
331	A Variance-Based Drift Metric for Inconsistency Estimation in Model Variant Sets	Nowadays, iterative development has become a state-of-the-art engineering process. The shared artifacts, i.e., models, must be edited collaboratively to make iterative development work in large engineering teams. The typical non-real-time collaboration workflow consists of three phases: The collaborators create copies of the model; Each collaborator edits their variant (copy) of the model; The collaborators merge the edited models back into one. During the merge phase, conflicting changes become apparent and must be resolved. This is a time and resource-intensive task. However, if potential merge conflicts can be detected during the editing phase, the collaborators can take suitable measures in time. This work proposes an early warning system for merge conflicts in model-based development projects. We introduce the novel metric Drift to quantify the inconsistency between all co-existing variants of a model. An increase in Drift indicates an increase in potential merge conflicts. We evaluate the correctness of the Drift for synthetical modeling projects and syntactical model differences. We develop an openly available tool for calculating the Drift for arbitrary Git repositories.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
332	A systematic mapping study on group work research in computing education projects	Context: For developing students' group-and teamwork skills needed in the team-oriented work environments of the software industry, the role of project-based learning is considered central. Yet there does not appear to be a proper mapping of the current group work research in the computer science project education literature. Thus, the current state of group work research in the research area is somewhat unknown.Objective: This study aims to form an overview of how research has addressed students' group work in the field of computer science (CS) and software engineering (SE) to identify research gaps as well as suitable topics for more detailed literature reviews.Methods: A systematic mapping study was used to investigate how group work in tertiary education has been undertaken in the literature during the past decade.Results: Based on the selected papers, the most investigated group work areas were related to the assessment of groups, group formation, communication, and cooperation. The research appeared to be quite narrowly focused on a few areas. Most of the papers were experience or evaluation research. A case study using interviews or questionnaires to gather data from a single course was the most representative type of study. The papers were mainly published in scientific conferences. The use of theoretical frameworks was limited, with a focus on a few established frameworks. Tuckman's group development theory was the predominant framework, while other commonly used concepts and theories include social loafing, Kolb's learning style theory, and the Big Five personality traits model.Conclusion: Out of 7515 papers screened, 225 were deemed eligible and analyzed. We conclude a need for more focused group work research in CS/SE student projects, in which education is inspected from particular perspectives. This would create identifiable lines of research and structure the research area. Relatedly, we suggest that the underused theoretical frameworks can inspire important research: group interventions would benefit from a socially shared regulated learning perspective, explicit use of justice theories would improve theoretical understandings of group behavior, and transactional distance would help analyze how students adopt a software process. Moreover, the research area could be precipitated by novel theoretical perspectives. For practitioners, those implementing a group project course can benefit from a large amount of literature on assessment and group formation, which are issues on which the teacher must take a position. We also include a lessons-learned summary for teachers. Generally, the present results outlining the field in a structured way can facilitate research-based teaching.& COPY; 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).	☑	☑	☐	☐	☐	☐	☐	☐
333	A Reflection on the Use of Systemic Thinking in Software Development	The research examines the value and potential usefulness of using systemic thinking, which looks at the inter-connectedness of things, to comprehend the complexities of software development projects and the technical and human factors involved. It considers two different aspects of systemic thinking - psychological and sociological - and posits that these can assist in understanding how software teams function and attain their objectives, as well as the goals of the entities for which they work. Our research aims to provide a novel contribution to the field by investigating the use of systemic thinking in software development teams and organizations. We evaluate the reliability and validity of the survey applied to different groups of relevant participants, relate our findings to existing literature, and identify the most representative factors of systemic thinking. Despite the popularity of various factors that fall under the umbrella of 'systems thinking', there is limited understanding of their effectiveness in improving organizational performance or productivity, particularly when it comes to psychological and sociological systemic factors. The relationship between the use of systems thinking and organizational performance is often based on anecdotal evidence, rather than the identification and application of specific factors. Our work emphasizes the importance of understanding and applying such factors in order to build a solid foundation for the effective use of system dynamics and systems thinking tools, which is crucial for software development teams.	☑	☐	☐	☐	☐	☐	☑	☐
334	A Median-Based Fuzzy Approach to Software Quality Evaluation	Software quality evaluation is a challenging task in software engineering. A new group decision-making evaluation model is presented in this work. The new model is based on the VIsekriterijumska optimizacija i KOmpromisno Resenje (VIKOR) technique, in which a group regret measurement and a group satisfaction measurement are provided to increase the number of reference criteria in the decision-making process. We choose the median to represent the center of the data. Based on this, an entropy-based weighting method is proposed and used to determine the weights of decision makers. A new normalized projection is explored to measure the closeness between two evaluation matrices in a Pythagorean fuzzy setting. Several experimental analyses demonstrate that the entropy-based weighting method developed in this study is superior to traditional weighting methods. The median-based data center provides support for stable decision outcomes. Four dynamic experiments are reported on in this paper: The first one shows that the decision results remain stable throughout the entire experimental range; the second one demonstrates that the proposed normalized projection measure outperforms traditional projection measure; the third one demonstrates that the newly developed VIKOR method outperforms the traditional VIKOR method; and the last one identifies the optimal range for the three parameters of the proposed comprehensive VIKOR model.	☑	☑	☐	☐	☐	☐	☐	☐
335	A Genetic Algorithm-Based Approach to Support Forming Multiple Scrum Project Teams	Forming effective teams is an essential but challenging task, especially for organizations that carry out multiple projects simultaneously, a problem known as the Multiple Team Formation (MTF) problem. The literature presents several solutions for the MTF problem, mostly modeling it as a search problem. However, the existing solutions are not suitable for Scrum projects. We addressed this gap by developing an approach composed of two main steps. First, we designed a Structured Task Model to support creating developers' profiles given their performance on past Scrum projects. Then, given a set of target projects' technology requirements and the available developers' profiles, we developed a Genetic Algorithm to form the teams for a set of target projects. We evaluated the proposed approach by comparing the teams formed by our approach with the ones formed by project managers from one organization. Our approach achieved 85% of precision when compared with the teams provided by the project managers who worked on the same target projects. We also recorded an acceptance rate of up to 75%. The significant value of precision achieved suggests that our approach can provide teams close to the project managers' expectations. In addition, our Structured Task Model offers a promising way to build technical profiles semi-automatically for Scrum developers. In future work, we intend to investigate how to complement the developers' profiles by using other types of attributes and knowledge sources.	☑	☑	☐	☐	☐	☐	☐	☐
336	A Distributed Software Project Management Framework	The software industry is enjoying the permeable and trans-border flow of software markets and can access resources from all corners of the world. Software engineers gain international work experience through a distributed working environment. It involves participation from individuals with different cultures, languages, and geographic time zones to work on a single project. In addition to providing global opportunities for software experts and businessmen, it also introduces new project management challenges. Barriers exist in trust, communication, monitoring, languages, cultures, and time zones. Distance mode management faces more technical challenges due to stakeholders' ambiguous understanding and various documentation. This study addresses an in-depth analysis of challenges and currently practicing methods. Moreover, a new virtual project management framework is proposed to minimize issues and maximize the virtual project management team's throughput. The framework is compared with commonly used methodologies by experts who have experience in global software project management, and the analysis is performed using the analytical hierarchy process. The evaluation matrix has shown that the proposed model is adequate for distance project management with better score in virtual scope and virtual management). Its excellency is in standard documentation practice, change management, and improving re-usability practice that will enhance business goals and stakeholder's satisfaction.	☑	☐	☐	☐	☐	☐	☑	☐

MSL: ETAPA 1 - TRIAGEM INICIAL			EXCLUÍDOS: 312						RESTANTES: 26	
IDENTIFICAÇÃO			EXC	MOTIVO						
ID	TÍTULO	RESUMO		DUP	CE1	CE2	CE3	CE4	CE5	CE6
337	A Data-Driven Approach to Team Formation in Software Engineering Based on Personality Traits	Collaboration among individuals with diverse skills and personalities is crucial to producing high-quality software. The success of any software project depends on the team's cohesive functionality and mutual complementation. This study introduces a data-centric methodology for forming Software Engineering (SE) teams centred around personality traits. Our study analysed data from an SE course where 157 students in 31 teams worked through four project phases and were evaluated based on deliverables and instructor feedback. Using the Five-Factor Model (FFM) and a variety of statistical tests, we determined that teams with higher levels of extraversion and conscientiousness, and lower neuroticism, consistently performed better. We examined team members' interactions and developed a predictive model using extreme gradient boosting. The model achieved a 74% accuracy rate in predicting inter-member satisfaction rankings. Through graphical explainability, the model underscored incompatibilities among members, notably those with differing levels of extraversion. Based on our findings, we introduce a team formation algorithm using Simulated Annealing (SA) built upon the insights derived from our predictive model and additional heuristics.	☑	☐	☐	☐	☐	☐	☐	☑
338	A case study in statistical software development for advanced evidence synthesis: the combined value of analysts and research software engineers	BackgroundSince 2015, the Complex Reviews Synthesis Unit (CRSU) has developed a suite of web-based applications (apps) that conduct complex evidence synthesis meta-analyses through point-and-click interfaces. This has been achieved in the R programming language by combining existing R packages that conduct meta-analysis with the shiny web-application package. The CRSU apps have evolved from two short-term student projects into a suite of eight apps that are used for more than 3,000 h per month.AimHere, we present our experience of developing production grade web-apps from the point-of-view of individuals trained primarily as statisticians rather than software developers in the hopes of encouraging and inspiring other groups to develop valuable open-source statistical software whilst also learning from our experiences.Key challengesWe discuss how we have addressed challenges to research software development such as responding to feedback from our real-world users to improve the CRSU apps, the implementation of software engineering principles into our app development process and gaining recognition for non-traditional research work within the academic environment.Future developmentsThe CRSU continues to seek funding opportunities both to maintain and further develop our shiny apps. We aim to increase our user base by implementing new features within the apps and building links with other groups developing complementary evidence synthesis tools.	☑	☐	☐	☐	☐	☐	☑	☐