CINTED-UFRGS Revista Novas Tecnologias na Educação

Software Engineering Topics in the Brazilian Undergraduate Programs: A Curricula Evolution Analysis

Jorge Marques Prates, UEMS, jprates@uems.br
https://orcid.org/0000-0002-6798-7263
Rogério Eduardo Garcia, UNESP, rogerio.garcia@unesp.br
https://orcid.org/0000-0003-1248-528X
José Carlos Maldonado, USP, jcmaldon@icmc.usp.br
https://orcid.org/0000-0002-3779-7143

Abstract: *Software engineering plays a vital role in Computer Science, with constantly evolving tools and methods. Curriculum guidelines from ACM and SBC help ensure programs align with industry demands. This study examines how Brazilian undergraduate Computer Science programs incorporate software engineering topics by analyzing university curricula, with a focus on students graduating in 2024. Core topics, such as software design, testing, and requirements engineering, are generally well-covered. Recent updates show growing adoption of agile development, cloud computing, and artificial intelligence. The inclusion of practical courses is becoming more common, enabling students to apply theoretical concepts in real-world scenarios and fostering alignment between academic training and professional expectations.*

Keywords: *Software Engineering Education, Teaching, Training, Curricula.*

Tópicos de Engenharia de Software nos Cursos de Graduação Brasileiros: Uma Análise da Evolução dos Currículos

Resumo: *A engenharia de software desempenha um papel essencial na Ciência da Computação. Diretrizes curriculares da ACM e da SBC ajudam a garantir o alinhamento dos cursos com as demandas do setor. Este estudo investiga a integração de tópicos de engenharia de software nos cursos de graduação em Ciência da Computação no Brasil, por meio da análise dos currículos acadêmicos, com ênfase nos estudantes que concluíram sua formação em 2024. Tópicos como projeto de software, testes e engenharia de requisitos são bem abordados. Atualizações indicam uma adoção de práticas como desenvolvimento ágil, computação em nuvem e inteligência artificial. Atividades práticas têm se tornado mais comuns, permitindo aos estudantes aplicar a teoria em cenários reais e promovendo uma maior aproximação entre a academia e o mercado profissional.*

Palavras-chave: *Educação em Engenharia de Software, Ensino, Treinamento, Currículos.*

1. Introduction

Software engineering, first introduced in the 1960s, applies a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software (IEEE..., 1990; PRESSMAN; MAXIM, 2019; KUMAR *et al.*, 2024). On the other hand, software engineering education has emerged from a series of courses offered by IBM (MEAD, 2008). Since then, the software engineering community has endeavored to organize existing knowledge in the field and transform it into a curriculum. As a result of these studies, curricula such as SWEBOK (BOURQUE; FAIRLEY, 2014), SEEK (SOBEL, 2003), and Computer Science Curricula (Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM); IEEE Computer Society,

2013) have emerged. Currently, software developers are trained through traditional education methods. However, this approach does not fully meet the current demands of the job market. Moreover, traditional education is not sufficient to equip students with up-to-date knowledge. Software engineering educators should prepare their students for various roles and help them stay aware of the rapid changes (SHAW, 2000; DEVADIGA, 2017).

Studies that investigate how much Computing/SE curricula meet the industry's expectations (MORENO *et al.*, 2012; RADERMACHER; WALIA, 2013) are found in the literature. For example, Liargkovas et al. (LIARGKOVAS *et al.*, 2022) manually analyzed 500 Wikipedia articles cited on Stack Overflow, systematically mapping them to the knowledge units defined in the SEEK model. The article highlights the gap between software engineering education and industry demands, with the SEEK model covering 79% of the topics cited on Stack Overflow but showing deficiencies in areas such as software components and specialized tools. The popularity of topics on Stack Overflow does not reflect their coverage in the curriculum, and programmers often seek practical guidance, especially in software design. Another important aspect in the teaching of software engineering is the need to develop practical activities, as it is an area that involves the application of methods, techniques, and tools for constructing a software product.

Developing curricular guidelines for software engineering is challenging, given the rapid evolution and expansion of new methods, tools, and practices in the field. Thus, the curricula of undergraduate programs must be appropriate to the demand for constant evolution. This work is inserted into this context, investigating whether these curricula meet the guidelines. This paper aims to examine how software engineering topics are integrated into the curricula of Computer Science undergraduate programs at Brazil's top universities for students entering the job market in 2024. Additionally, the study was replicated to analyze the evolution of curricula and the development of graduates' skills by 2024. This research establishes a baseline for curriculum design and graduate training as a reference point for future comparisons and identifying significant changes over time.

The universities were selected according to the 2018 RUF ranking (Ranking of Universities *Folha*). After the selection, the curricula of the Computer Science undergraduate programs, specifically those concerning software engineering courses, were analyzed. The results indicate that the curricula of the 2018 undergraduate programs generally cover the suggested software engineering topics. However, theoretical lectures dominate, often comprising up to 100% of the regular hours, with limited practical hours. In contrast, the 2024 curricula emphasize hands-on learning and adapt to emerging practices such as agile development, DevOps, and AI in software engineering. While the overall structure remains consistent, universities are incorporating modern methodologies to meet the needs of the industry. The addition of specialized electives and a focus on automated development processes enhances the curriculum's flexibility.

The remainder of this paper is organized as follows. Section 2 presents the background of software engineering curricula, especially the computing curricula guidelines and SBC undergraduate computing curriculum. The research method used for curricula data extraction is described in Section 3. The results regarding the universities' curricula obtained from this study are presented and discussed in Section 4. Finally, we present conclusions for this research in Section 5.

2. Related Work

Software engineering comprises software development tools, methods, resources, and technologies. Besides, it is required that the curricula be evolved enough to meet the area evolution and adopt a project-centric curriculum (GARY *et al.*, 2013). In this context, its teaching is in the curriculum guidelines of computing undergraduate programs, such as Computer Science, Computer Engineering, and Information Systems.

The Software Engineering Body of Knowledge (SWEBOK) serves as a guide for utilizing and applying best software engineering practices. In this guide, the IEEE Computer Society establishes a base for the body of knowledge in the field of software engineering, promoting the advancement of theory and practice in this area. The SWEBOK version 3, released in 2014, was a guiding framework for software engineering practices for ten years. In 2024, version 4.0 was introduced, featuring new topics that address and integrate emerging areas in software engineering, reflecting the field's continuous evolution and advancements. The latest version of the SWEBOK introduces comprehensive updates to all Knowledge Areas (KAs) to reflect advancements in software engineering, including modern development practices and innovative techniques. Agile and DevOps methodologies have been extensively incorporated across the KAs due to their widespread adoption. Additionally, emerging technologies such as artificial intelligence (AI), machine learning (ML), and the Internet of Things (IoT) have been integrated (WASHIZAKI, 2024).

Another set of guidelines can be found at SEEK (Software Engineering Education Knowledge), developed by the ACM Education Board and the IEEE-Computer Society Educational Activities Board, aimed at creating curriculum recommendations for various computing disciplines, including computer science, computer engineering, software engineering, and information systems (SOBEL, 2003). Its structure is composed of areas that were selected according to SWEBOK KAs. Furthermore, it is based on various discussions with the software engineering community. A set of results for the undergraduate curriculum in software engineering is also defined. The guide defines that students are expected to be able to work as part of a team to develop and deliver executable artifacts, understand the process of determining customers' needs and translating them into software requirements, reconcile conflicting objectives, and find acceptable solutions within the constraints of cost, time, knowledge, existing systems, and organizations.

The Computer Science Curricula (CSC) includes a specification of the content to be covered and a curriculum that implements these specifications, both of which are essential components necessary for a comprehensive Computer Science curriculum (Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM); IEEE Computer Society, 2013). According to the CSC, software engineering is the discipline focused on applying theory, knowledge, and practice to build reliable software systems effectively and efficiently, ensuring that these systems meet the requirements of both customers and users. Students and instructors must understand how specialization impacts software engineering approaches to achieve this. The 2023 version of the CSC introduces changes to the Software Engineering Knowledge Area compared to the objectives set in CS2013. The main updates include a focus on the software workflow, covering version control, testing, code review, and tools, as well as an increased emphasis on soft skills such as team participation, communication, and collaboration (KUMAR *et al.*, 2024).

The Brazilian Computing Society (SBC) proposed the reference curricula for undergraduate computing programs in Brazil. According to the 2005 SBC guidelines, an undergraduate program that includes a course on software engineering in its curriculum should cover the following topics (SBC., 2005a; SBC., 2005b): (1) Software Development Process; (2) Software Development Life Cycle; (3) Software Quality; (4) Software Planning and Management Techniques; (5) Software Configuration Management; (6) Requirements Engineering; (7) Systems Analysis and Design Methods; (8) Software Quality Assurance; (9) Verification (i), Validation (ii) and Software Testing (iii); (10) Software Maintenance; (11) Documentation; (12) Design Patterns; (13) Software Reuse; (14) Reverse Engineering; (15) Reengineering; (16) Software Development Environment.

In 2017, the SBC published a book offering curricula recommendations for Computing programs, including Computer Science, Computer Engineering, Software Engineering, Computing Education, Information Systems, and higher education technology courses in Computing (ZORZO *et al.*, 2017). According to these curricula, Computer Science graduates are expected to be professionals who have the opportunity to develop key aspects related to their academic training, such as teamwork, innovation, and adapting to technological changes. Acting with professionals from diverse fields to identify opportunities and address societal needs requires strong teamwork skills. Training in this context fosters an understanding of the world and society, emphasizing the development of collaboration, communication, and expression. Additionally, it involves business-oriented education that provides insight into organizational dynamics and a commitment to staying updated with technological advancements and the state of the art.

3. Methodology

The first step of the applied methodology involved defining the research design, which incorporated elements of Qualitative Research Methods (TAYLOR; BOGDAN; DEVAULT, 2016). During this phase, key aspects, including data sources, document types, data collection procedures, and data preparation and analysis processes, were established.

The definition of the data source involved selecting Brazilian universities whose curricula were analyzed. This selection was based on the 2018[1] ranking published by RUF (Ranking of Universities *Folha*). RUF, an annual evaluation of higher education institutions in Brazil conducted by the Folha de S. Paulo, a newspaper that has been assessing universities based on key indicators since 2012. In the 2018 ranking, universities were evaluated according to five criteria: research, internationalization, innovation, education, and job market performance. Similarly, subsequent rankings maintained these research-based indicators. The ranking of 2018 is shown in Table 1.

In the 2018 edition of the RUF, the ranking of undergraduate Computer Science programs was published, in addition to the ranking of universities. In this list of programs, you can find evaluations of Brazil's most popular undergraduate programs, which are based on five indicators: research, internationalization, innovation, education, and the job market. Each classification considered the undergraduate programs offered by universities, university centers, and colleges. For this research, Brazil's main undergraduate Computing programs were selected, shown in Table 2.

The documents were extracted at two distinct points, specifically in 2018 and 2024, to facilitate a comparative analysis of the data from these periods. The extracted documents were texts that described the curricula of Computer Science courses. The data collection was performed according to the following procedure: (i) visiting the websites

[1] ⟨https://ruf.folha.uol.com.br/2018/ranking-de-universidades/⟩

Tabela 1. Top-ranked Brazilian universities in 2018

Brazilian Universities	Abbreviation
University of São Paulo	USP
Federal University of Rio de Janeiro	UFRJ
Federal University of Minas Gerais	UFMG
State University of Campinas	UNICAMP
Federal University of Rio Grande do Sul	UFRGS
Federal University of Santa Catarina	UFSC
Federal University of Paraná	UFPR
São Paulo State University	UNESP
University of Brasilia	UNB
Federal University of Pernambuco	UFPE
Federal University of São Carlos	UFSCAR
Federal University of Ceará	UFC
State University of Rio de Janeiro	UERJ
Federal University of Bahia	UFBA
Federal University of Viçosa	UFV
Federal University of Fluminense	UFF
Federal University of São Paulo	UNIFESP
Pontifical Catholic University of Rio Grande do Sul	PUC-RS
Pontifical Catholic University of Rio de Janeiro	PUC-RIO
Federal University of Goiás	UFG
Federal University of Santa Maria	UFSM
Federal University of Rio Grande do Norte	UFRN
Federal University of Juiz de Fora	UFJF
State University of Londrina	UEL
State University of Maringá	UEM

of each Higher Education Institution in RUF; (ii) accessing the list of undergraduate programs offered by the Higher Education Institution; (iii) selecting the Computer Science undergraduate program and, if available, accessing the undergraduate program website; (iv) in cases where more than one Computer Science undergraduate program was found at the university, the earliest one was selected; (v) if available, the curricula or the pedagogical projects were downloaded for future analysis.

The data preparation involved reading documents containing information on software engineering disciplines and related disciplines such as software testing, project management, and software architecture. The following elements were extracted with a data extraction form: discipline title, covered topics, and required and elective hours. Finally, the data were analyzed using the reference curricula available by SBC. The discussion and results are presented in the next section.

4. Results and Discussion

The first part of the results focuses on data from the 2018 curricula of the top 25 universities in Brazil and the top 25 Computer Science undergraduate programs, as ranked by the RUF. It is important to highlight that most students who completed their studies under these curricula transitioned into the job market in 2024, likely benefiting from the skills and knowledge acquired during their education at these institutions. The analysis, summarized in Table 3, highlights that the top 10 most reputable universities also have Computer Science undergraduate programs ranking among the highest positions.

In addition, Table 3 includes information about the topics covered by the university's curricula that relate to the SBC curriculum topics. Notably, the table excludes data from institutions whose curricula were inaccessible via official websites and from those that did not reply to email inquiries, despite repeated follow-up attempts. This information enables us to assess the coverage of the topics and determine whether each university undergraduate program addresses the suggested topics. Another information in the table is the existence of Postgraduate programs in computing in the top-ranked universities (Mas-

Tabela 2. Top-ranked Brazilian universities in Computer Science in 2018

Brazilian Universities	Abbreviation
State University of Campinas	UNICAMP
University of Sao Paulo	USP
Federal University of Rio Grande do Sul	UFRGS
Federal University of Minas Gerais	UFMG
Federal University of Rio de Janeiro	UFRJ
Federal University of Pernambuco	UFPE
Federal University of Santa Catarina	UFSC
Federal University of São Carlos	UFSCAR
Pontifical Catholic University of Rio de Janeiro	PUC-RIO
University of Brasilia	UNB
Federal University of Paraná	UFPR
São Paulo State University	UNESP
Pontifical Catholic University of Rio Grande do Sul	PUC-RS
Pontifical Catholic University of Paraná	PUC-PR
Federal University of Campina Grande	UFCG
Federal University of Technology - Paraná	UTFPR
Federal University of Amazonas	UFAM
Federal University of Ceará	UFC
Pontifical Catholic University of Minas Gerais	PUC MINAS
Pontifical Catholic University of São Paulo	PUC-SP
Federal University of Santa Maria	UFSM
Federal University of Sergipe	UFS
Federal University of Goiás	UFG
University of Vale do Rio dos Sinos	UNISINOS
Federal University of Pará	UFPA

ter's and Ph.D.). This analysis is particularly relevant as the existence of these programs often reflects the maturity and academic depth of the associated undergraduate courses.

Tabela 3. Curricula Mapping of the Top-Ranked Universities and Computer Science Undergraduate Programs in 2018

University	Rankings - 2018		Postgraduate Program		Summary Topics															
	General	Computer Science	Ms	PhD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
USP	1	2	X	X	X	X	X	X	X	X	X	X	X	X		X	EC			X
UFRJ	2	5	X	X	X	X	X	X	X	X	X	X	X			X				X
UFMG	3	4	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	
UNICAMP	4	1	X	X	X	X	X	X	X	X	X	X	X	X		X		X	X	X
UFRGS	5	3	X	X	X	X		X		X	X		X	X		X				X
UFSC	6	7	X	X	X	X	X	X		X	X	X	X	X		X	X	X	X	X
UFPR	7	11	X	X	X	X	X	X		X	X	X	X			X				
UNESP	8	12	X	X	X	X	X	X	X	X	X	X	X	X		X				X
UNB	9	10	X	X	X	X	X	X			X	X	X	X						X
UFPE	10	6	X	X	X	X	X	X	X	X	X	X	X	X		X		X	X	X
UFSCAR	11	8	X	X	X	X	X	X		X	X	X	X			X		X	X	X
UFC	12	18	X	X			X	X	X	X	X	X	X	X	X					
UERJ	13	29	X		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFBA	14	27	X	X	X	X	X	X		X	X	X	X		X	X				X
UFV	15	47	X		X	X	X	X		X										
UFF	16	26	X	X	X	X	X	X	X	X	X	X	X			X		X	X	X
UNIFESP	17	58	X	X	X	X		X	X	X	X		X	X						X
PUC-RS	18	13	X	X	X	X				X	X		X							X
PUC-RIO	19	9	X	X			EC			X	X	EC	X			X				X
UFG	20	23	X	X	X	X	X	X	X	X	X	X	X	X		X				X
UFSM	21	21	X		X	X				X	X		X			X				X
UFRN	22	28	X	X	X	X			X		X			X	X				X	
UFJF	23	41	X	X	X		X	X	X	X		X	X	X						X
UEL	24	51	X		X	X	X	X		X	X	X	X	X		X				X
UEM	25	34	X		X	X	X	X	X	X	X	X	X	X	X	X	X			X
PUC-PR	31	14	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFCG	41	15	X	X	X	X	X	X		X	X	X	X	X		X				X
UTFPR	55	16	X		X	X	X	X	X	X	X	X	X	X						X
UFAM	47	17	X	X	X	X	X	X		X	X	X	X	X		X				EC
PUC-MINAS	56	19	X	X	X	X	X	X	X	X	X		X			X				X
PUC-SP	54	20			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFS	38	22	X		X	X	X	X	X	X	X	X	X	X		X	EC	X	X	X
UNISINOS	36	24	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFPA	25	27	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

None of the 16 topics were covered 100%. However, topics addressing fundamental software engineering concepts, such as development processes and life cycles, requirements engineering, analysis and design methods, and software testing, were covered in almost complete detail. In summary, the analyzed courses addressed the core principles of software engineering, providing students with valuable knowledge in these foundational areas. On the other hand, only some undergraduate programs addressed their curricula topics, such as software reuse, reverse engineering, and reengineering. Documentation is a topic that can be implicit in the process of Software Engineering learning, which is why few Computer Science undergraduate programs address this topic.

Based on the available data, the study categorized the curricula hours as required or elective. However, some programs did not specify whether the courses were elective or mandatory. Additionally, several programs did not distinguish between theoretical and practical lecture hours, which made it challenging to fully assess the depth and nature of the content covered in these courses. A notable observation was that nearly half of the programs offered software engineering courses with a theoretical focus only. On the other hand, the remaining programs exhibited a more balanced distribution between theoretical and practical lecture hours. These programs were likely better aligned with industry needs, incorporating both conceptual understanding and practical application.

In addition to the core topics, some regular courses included subjects such as Agile Software Development, Web Techniques, and Experimental Software Engineering, which reflected industry trends in 2018 and supported students' adaptation to modern practices. Elective courses, in turn, offered a broader range of topics—such as Software Engineering for Embedded Systems, Educational Software Engineering, and Formal Specification of Software—highlighting the flexibility of the curricula and allowing students to specialize according to specific career paths or research interests.

The second part of the analysis focuses on the curricular content of the same universities in 2024, as presented in Table 4. A comparison with the 2018 curricula reveals no significant changes in these topics over the years. However, beyond these core topics, the new course syllabus has incorporated agile development education, reflecting the growing adoption of these practices in the software industry, including Scrum, user stories, and agile project management. New specialized Software Engineering courses have been introduced, including a fully practical course focused on integrating development, validation, and verification (V&V) tools in building information systems.

Overall, the new curricular structures are organized into specialized Software Engineering courses, covering different stages of the development cycle, such as Software Systems Analysis, Software Systems Design, and Software Systems Management. Another significant aspect is the increased focus on practical courses, in which students consolidate their knowledge through real-world projects developed in partnership with organizations. In these courses, students work in teams under the guidance of a professor, applying the concepts acquired throughout the program. Furthermore, there is an emphasis on using CASE tools during the design phase, providing an experience that closely aligns with current market demands and prepares students for professional challenges.

The innovations in the curricula include the automation of the software development process, covering environments based on operating systems and programming languages, as well as modern methodologies. Automation is enhanced by artificial intelligence techniques, including intelligent tools that assist in software production. In addition, software project management techniques, both local and geographically distributed, are evolving with the adoption of new approaches that facilitate coordination and

Tabela 4. Curricula Mapping of the Top-Ranked Universities and Computer Science Undergraduate Programs in 2024

University	Rankings - 2024		Postgraduate Program		Summary Topics															
	General	Computer Science	Ms	PhD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
USP	1	2	X	X	X	X	X	X	X	X	X	X	X	X		X	EC			X
UFRJ	4	5	X	X	X	X	X	X	X	X	X	X	X			X				X
UFMG	5	3	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	
UNICAMP	2	1	X	X	X	X	X	X		X	X	X	X			X				
UFRGS	3	6	X	X	X	X		EC		X	X		X	X	X					
UFSC	7	7	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X
UFPR	9	14	X	X	X	X	X	X		X	X	X	X			X				
UNESP	6	9	X	X	X	X	X	X	X	X		X	X	X		X				X
UNB	8	12	X	X	X	X	X	X		X	X	X	X	X	X					X
UFPE	11	4	X	X	X	X	X	X		X	X	X	X			X		X	X	X
UFSCAR	12	8	X	X	X	X	EC	EC		X	X	EC	EC	EC						
UFC	20	13	X	X			X	X	X	X	X	X	X	X	X					
UERJ	21	77	X	X	X	X	X	X	X	X		X	X	X	X					X
UFBA	18	11	X	X	X	X	X	X		X	X	X	X		X	X				X
UFV	15	56	X	X	X	X		X		X	X		X			EC				
UFF	17	17	X	X	X	X	X	X		X	X	X	X	X		X		X	X	
UNIFESP	10	29	X	X	X	X		X	X	X	X		X	X		X	X			X
PUC-RS	22	18	X	X	X	X			X	X	X		X							X
PUC-RIO	25	15	X	X			X	X				X	X			X				X
UFG	19	69	X	X	X	X	X	X	X	X	X		X	X		EC	EC	EC	EC	
UFSM	13	76	X		X	X	X	X	X	X	X	X	X		X	X				X
UFRN	16	27	X	X	X	X	X	X	EC	X	X	EC	X	X		X				
UFJF	30	74	X	X	X		X	X	X	X		EC	X	EC		EC		EC	EC	
UEL	27	70	X		X	X	X	X		X	X	X	X	X						X
UEM	24	26	X	X	X	X	X	X		X	X	X	X		X	X	X			
PUC-PR	32	21	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFCG	49	19	X	X	X			X		X	X	X	X		EC	X				
UTFPR	35	10	X		X	X	X	X	X	X	X	X	X	X		X	X			X
UFAM	43	24	X	X	X	X	X	X		X	X	X	X	X		X				
PUC-MINAS	63	22	X	X	X		X	X	X	X	X	X	X			X	X			
PUC-SP	64	32			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFS	29	25	X		X	X	X	X	X	X	X	X	X	X		X	EC	X	X	X
UNISINOS	34	39	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UFPA	31	31	X	X	X	X	-	X	X	X	X	-	X	X	X	-	-	-	-	-

collaboration among teams in different locations.

In terms of specific disciplines, areas such as Empirical and Experimental Software Engineering, as well as Economics for Software Engineering, stand out. Another significant advancement is Model-Driven Software Engineering, which focuses on model-driven architectures and transformations between models. This approach involves the creation of platform-independent and platform-dependent models, the generation of models from other models, the creation of code from models, and even the creation of models and code from code. Model property verification also plays a crucial role in this context.

The Software Product Line Development process is being addressed in more detail, focusing on Domain and Application Engineering. The automatic derivation of software products is another critical point, allowing the creation of distinct product versions from a single code repository. Strategies for adopting Software Product Lines, such as Extractive, Reactive, and Proactive approaches, are being incorporated, providing a broader understanding of how organizations can adopt this methodology. Additionally, techniques for implementing variations in software product lines are being introduced.

Specific courses also cover in-depth studies on emerging topics in Software Engineering. The content of these courses explores new trends, including advanced and current topics such as approaches related to Artificial Intelligence. Among the issues discussed are the use of GenAI and LLMs in Software Engineering and emerging practices such as prompt engineering. These courses aim to equip students with knowledge of the latest innovations, preparing them to tackle challenges and capitalize on the opportunities presented by AI in software development. Another emerging trend is cloud software development and DevOps, which emphasize continuous software delivery and a solid understanding of key concepts such as the DevOps pipeline. The goal is to integrate tools and methodologies to build an efficient pipeline that includes version control, continuous integration, and continuous deployment. These principles foster automation and enhance development efficiency, enabling more agile workflows better aligned with market demands.

Some special topics courses cover Software Engineering themes that are not addressed in other curricular components. Examples of these elective courses include Software Development for Web and Mobile Platforms, Microservices Architecture, Aspect-Oriented Software Engineering, Model-Driven Software Engineering, Search-Based Software Engineering, Embedded Software Engineering, and Test Automation with Continuous Integration Tools. The inclusion of concepts related to Introduction to Software Product Lines in the new elective syllabus was also observed. This topic covers the field's fundamental concepts, such as characteristics, variation points, and variations. Furthermore, the syllabus addresses Variability Modeling and Management, an essential aspect for managing different software versions and configurations.

5. Final Remarks

This study investigates how software engineering is addressed in undergraduate Computer Science programs at Brazil's top-ranked universities. These programs were analyzed based on the curriculum guidelines proposed by the SBC, which serve as a national reference for software engineering education. Focusing on the training received by students who entered the job market in 2024, the research highlights key elements of their educational background and their alignment with industry expectations.

In 2018, most curricula consisted primarily of theoretical courses, many of which were entirely theoretical. However, given the practical nature of software engineering, hands-on experience with tools and real-world activities is essential. The lack of practical components can be partly attributed to the extensive range of topics that need to be addressed within the limited average course duration of about 120 hours. In the 2024 curricula, this limitation was more effectively mitigated through the introduction of system development-focused courses.

By examining how software engineering topics are integrated across different institutions, the study provides valuable insights into how well students are being prepared to meet the demands of professional practice, such as meeting project deadlines, working in teams, and adapting to rapidly evolving technologies. The research emphasizes the importance of hands-on learning and equipping students with practical skills to respond to these dynamic environments. Furthermore, the study reveals a clear evolution in curricula, marked by the adoption of emerging practices such as agile development, DevOps, cloud computing, and artificial intelligence. Despite these advances, certain limitations affect the validity of the study. Some curricula available on university websites may be outdated, and difficulties in accessing complete and up-to-date information, especially due to the lack of clarity on where to find the program documents, occasionally hinder data collection. To address this issue, additional information was requested directly from program coordinators.

Overall, the results suggest that while the foundational structure of software engineering education remains consistent, universities are increasingly embracing modern methodologies to meet industry needs. The addition of specialized electives and a stronger emphasis on practical development ensures greater curricular flexibility and relevance.

Referências

BOURQUE, P.; FAIRLEY, R. E. (Ed.). *SWEBOK: Guide to the Software Engineering Body of Knowledge*. Version 3.0. Los Alamitos, CA: IEEE Computer Society, 2014.

DEVADIGA, N. M. Software engineering education: Converging with the startup industry. In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, 2017. p. 192–196.

GARY, K. *et al.* A project spine for software engineering curricular design. In: *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, 2013. p. 299–303.

IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, p. 1–84, Dec 1990.

Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM); IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: Association for Computing Machinery, 2013.

KUMAR, A. N. *et al. Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024.

LIARGKOVAS, G. *et al.* Software engineering education knowledge versus industrial needs. *IEEE Transactions on Education*, v. 65, n. 3, p. 419–427, 2022.

MEAD, N. R. Software engineering education: How far we've come and how far we have to go. In: *2008 21st Conference on Software Engineering Education and Training (CSEE&T)*, 2008. p. 18–22.

MORENO, A. M. *et al.* Balancing software engineering education and industrial needs. *Journal of Systems and Software*, v. 85, n. 7, p. 1607–1620, 2012.

PRESSMAN, R. S.; MAXIM, B. R. *Software Engineering: A Practitioner's Approach*. 9. ed. New York, NY, USA: McGraw-Hill, Inc., 2019.

RADERMACHER, A.; WALIA, G. Gaps between industry expectations and the abilities of graduates. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*. New York, NY, USA: ACM, 2013. v. 1, p. 525–530.

SBC. *Currículo de Referência da SBC para Cursos de Graducão em Bacharelado em Ciência da Computação e Engenharia de Computação, 2005*. 2005.

SBC. *Currículo de Referência da SBC para Cursos de Graducão em Computação e Informática, 2005*. 2005.

SHAW, M. Software engineering education: a roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2000. (ICSE '00), p. 371–380.

SOBEL, A. E. K. *Computing Curricula – Software Engineering Volume – Final Draft of the Software Engineering Education Knowledge (SEEK)*, 2003.

TAYLOR, S. J.; BOGDAN, R.; DEVAULT, M. L. *Introduction to Qualitative Research Methods: A Guidebook and Resource*: JohnWiley & Sons, 2016.

WASHIZAKI, H. (Ed.). *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*. Version 4.0: IEEE Computer Society, 2024.

ZORZO, A. F. *et al. Referenciais de Formação para os Cursos de Graduação em Computação*: Sociedade Brasileira de Computação (SBC), 2017. 153 p.