The application of continuous practices in higher computer science education - A systematic literature review

Florian Huber
Faculty Computer Science
Kempten University of Applied Sciences
Kempten, Germany
florian.huber@hs-kempten.de

Georg Hagel
Faculty Computer Science
Kempten University of Applied Sciences
Kempten, Germany
georg.hagel@hs-kempten.de

***Abstract*—In the field of computer science, Continuous Practices enable companies to frequently and instantly provide new software and products to customers and stakeholders. With a growing interest in these practices, some secondary literature has been published within this research area. However, there are still open questions when it comes to teaching such practices to computer science students. With more and more companies demanding these skills from graduated students, educating them the required knowledge and skills is necessary. This systematic literature review follows the methodology of Kitchenham and analyses which of these practices are taught in higher computer science education. Along with the kind of courses that use them, it is reviewed how they are taught in higher computer science education and how these approaches differ from each other. The systematic literature review points out, that there are currently different teaching approaches described in literature. The review might help educators to gain new ideas of how to develop an own course to teach such practices or how to implement such contents into existing courses.**

***Index Terms*—Continuous Practices, Software Engineering, Higher Education, Systematic Literature Review**

I. Introduction

In recent years, Continuous Practices (CP) have gained much interest from both, industry and education [1], [2]. For companies, CP proved to be very advantageous. During the software development process, they can deliver instant feedback for written code. Furthermore, a compiled and tested executable can be provided automatically. CP enable companies to deploy new features and react to changed environment conditions instantly and on a daily basis.

Companies seem to have great interest in employees, which have skills in the field of CP. This motivates educators to teach them to their students. However, setting up these environments is a complex challenge [3], [4]. Therefore, a thoughtful teaching approach on the part of educators is required. Alongside the implementation, teaching the theoretical aspects of CP comes with challenges for students and educators as well [5].

The contribution of this work will be a systematic literature review (SLR) for a better understanding about the current state of published research in this field of teaching. Also it will be evaluated how CP are used and taught in higher computer science education and how these approaches differ from each other. For educators, this review might deliver a deeper understanding of the different ways to teach them and how they could be integrated into their own curricula.

It is important to mention, that this SLR is based on current publications within this field of research. All analysis and statements refer to these. This publication does not cover a study about unpublished teaching practices at different universities. Further studies about how CP are taught in practice (without being published) might be covered in future work.

The rest of this paper has the following structure: section II discusses related work. The research method used in this paper is outlined in section III. Finally, the results of this SLR are presented in section IV.

II. Related work

Along with primary publications, some secondary literature was developed in this active topic of research. This section will enumerate related secondary literature and distinguish it from the SLR developed in this paper.

Because of its high relevance, there are quite a few secondary papers about DevOps. The authors in [6], for example, perform a systematic mapping study to explore primary literature about it. In this study they point out, that DevOps has not been sufficient studied so far. [7] analyses published literature about CP and DevOps. The authors verify how field related terms are used and distinguish them from each other. Authors in [8] investigate, which contributions have been made by researchers in primary literature to improve DevOps processes.

Besides DevOps itself, also its cultural aspects have been analysed in secondary literature. [9], e.g., reviewed the invaluable cooperation and knowledge sharing between development (Dev) and operations (Ops). [10] examine three possible areas for changes in system design and architecture to better support CP. Authors in [11] empirically investigate how development and operations teams are organized in industry. Also they examine impacts of practicing CP on collaboration and the responsibilities of different team members.

Security must also be taken into account when developing software with DevOps. [12] gives an overview about security in the field of DevOps and what implementing it means. Authors in [13] performed a SLR to identify hurdles and develop a prioritization based taxonomy of DevOps security challenges.

Because of their relevance for DevOps, there is also secondary literature about CP. Authors in [3] review approaches, tools, challenges and practices alongside with success factors for applying CP.

Other researchers only focus on specific practices contained in CP. [14] show different interpretations and implementations of Continuous Integration (CI) in several cases. Another publication analyses how organisations assimilate CI and what organisational changes they may cause [15]. [16] conducts a systematic mapping study to classify and analyse literature about continuous deployment.

No matter if in industry or education, implementing CP is challenging and time consuming. This is also widely discussed in literature. For example, problems like the effect of build waiting time in CI [17] or which pain points occur when it comes to CI [18] are discussed. There is also secondary literature about testing and solutions for testing in Continuous Delivery (CDE). Publication [19] investigates approaches, methods, frameworks and solutions for testing problems.

How change management practices are adopted in environments that use CDE and Continuous Deployment (CD) is reviewed in [20]. [4] analyse problems, causes and solutions when it comes to adopting CDE.

[21] conducts a review about DevOps challenges for engineers, managers and researchers. Authors in [22] develop a model for DevOps to improve the adoption of its activities.

The researchers in [23] even investigated the customer involvement in the CD process, including potential benefits, challenges, methods and tools.

Only one secondary paper has been found, whose focus is on teaching such practices in an educational context. [24] identifies challenges and also recommendations for DevOps in education. To do so, the authors performed a systematic literature review and extracted 75 challenges and 83 recommendations, which they cluster into different categories. It is also discussed how challenges are tackled by existing recommendations. With this focus, the paper does not examine which CP are applied in educational environments. It should also be of interest for educators, how different approaches are applied and how they vary from each other. The insights of these, so far, uncovered topics will be the contribution of this paper.

III. Research method

The methodological basis of this review is based on Kitchenham [25], [26]. In the first phase, a review protocol shall be created. Following [26], the structured approach is illustrated by the following subsections.

TABLE I: Inclusion criteria of this SLR

ID	Inclusion criteria
I1	Papers from scholarly research journals and conference proceedings which are peer-reviewed.
I2	Papers need to be available in full-text and in German or English language.
I3	Papers need to be related to the field of computer science.
I4	The publication date of the paper must be in the period between January 1994 and October 2020.
I5	Papers which use CP in a higher (university) computer science education context.

A. Research questions

The following research questions are intended to be answered:

- **RQ1:** What CP are taught in higher computer science education, according to scientific literature?
- **RQ2:** In which courses and on which degree level (undergraduate or graduate) are CP taught to students, according to scientific literature?
- **RQ3:** How are CP taught in higher computer science education, according to scientific literature?

B. Electronic data sources

Following [27], the most popular electronic data sources in the field of software and software engineering are:

- IEEExplore (https://ieeexplore.ieee.org)
- ACM Digital Library (https://dl.acm.org/)
- ScienceDirect (https://www.sciencedirect.com/)
- Wiley Online Library (https://onlinelibrary.wiley.com/)
- SpringerLink (https://link.springer.com/)

Citing [28], Kitchenham and Charters [26] name the same electronic data sources. Because these databases are outlined by different authors it can be assumed, that they contain important primary literature for this SLR. Because the authors do not have access to the Scopus digital library (https://www.scopus.com), it is not included.

C. Inclusion and exclusion criteria

Inclusion and exclusion criteria for this SLR are summarized in the tables I and II. These criteria were developed by the authors, following the methodology and suggestions of Kitchenham [25], [26].

According to [7], the exact date of existence of CP is uncertain. The earliest date they mention is 1994 when Grady Booch formed the term of continuous integration [7], [29]. To be able to find all important literature for this review, articles from 1994 until October 2020 will be in focus of the literature search.

Literature which is not available in full-text and German (which is the mother tongue of the authors) or English language has been excluded. Otherwise, the authors would not be able to make valid statements about their content.

TABLE II: Exclusion criteria of this SLR

ID	Exclusion criteria
E1	Secondary literature.
E2	Papers which name any kind of CP, but do not describe how they are used to teach students.
E3	Work-In-Progress papers are only included, when already a concrete concept of how to teach CP or use CP in education exists.
E4	All papers which are not related to the field of computer science.

D. Primary study selection

1) Search term: To find primary studies in the named databases (see subsection: III-B) a search term has been developed and applied. However, it is not possible to apply the exact same search string to different databases (e.g. [28]). As a consequence it is necessary to slightly adapt the search string for each database. The following search term consists of synonyms for continuous practices (based on [3], filtered and enriched by the authors):

TITLE-ABS-KEY *(("DevOps"* ***OR*** *"continuous software engineering"* ***OR*** *"continuous practices"* ***OR*** *"continuous integration"* ***OR*** *"rapid integration"* ***OR*** *"fast integration"* ***OR*** *"quick integration"* ***OR*** *"frequent integration"* ***OR*** *"continuous delivery"* ***OR*** *"rapid delivery"* ***OR*** *"fast delivery"* ***OR*** *"quick delivery"* ***OR*** *"frequent delivery"* ***OR*** *"continuous deployment"* ***OR*** *"rapid deployment"* ***OR*** *"fast deployment"* ***OR*** *"quick deployment"* ***OR*** *"frequent deployment"* ***OR*** *"continuous release"* ***OR*** *"rapid release"* ***OR*** *"fast release"* ***OR*** *"quick release"* ***OR*** *"frequent release"* ***OR*** *"continuous build") **AND** ("education"* ***OR*** *"teach*"* ***OR*** *"student"* ***OR*** *"course") **AND**("software"* ***OR*** *"information systems"* ***OR*** *"computer"))*

If the search term contained to many operators, it has been split and the individual results compiled.

To verify if the search term is effective and no relevant paper had been excluded, a Google Scholar search was conducted as well, using different combinations of the keywords above. However, this did not deliver any further results.

2) Procedure: The search procedure is split into five phases. During the initial search in **phase 1**, *N = 2817* papers have been found. A vast number (2028 papers) on SpringerLink. It can be assumed, that this is due to the issue, that SpringerLink does not provide an opportunity to search by title, keywords etc. **Phase 2** applied the inclusion and exclusion criteria to title and keywords of the results from phase 1. After this step, *N = 61* papers were taken into consideration. Afterwards, in **phase 3**, the inclusion and exclusion criteria have been applied to the abstracts and conclusion sections of the papers. This resulted in *N = 45* papers to be left. In **phase 4**, following the methodology of [30] a snowballing process with a backward search has been used to find additional work related to this review. **Phase 5** covers a full text read of the remaining papers. A publication was chosen if it matched all the inclusion criteria after the full text read. A number of *N = 35* publications were selected within this process and will be synthesized in the following.

TABLE III: Distribution of different CP in the selected publications

Kind of Continuous Practices	Count	Percent
Continuous Integration	**32**	**91.43**
Continuous Delivery	**17**	**48.57**
Continuous Deployment	**8**	**22.86**
DevOps	**7**	**20.00**
DevSecOps	**1**	**2.86**

IV. Results

According to [3], reporting information about the distribution of selected papers in databases can be relevant to researchers, which are interested in this topic. Most of them have been found in the ACM Digital Library (40.00%) and IEEExplore (37.14%) databases. The others are distributed as follows: SpringerLink 20.00%, Wiley Online Library 2.86%. From Science Direct, no publication matched the chosen criteria.

A. RQ1: What CP are taught in higher computer science education, according to scientific literature?

Table III provides an overview about how many of the selected publications use different CP. With the total values, a percentage distribution can be calculated (the total of 35 selected papers equals 100%).

Nearly all of them (91.43%) use CI in an educational context. Whereby a smaller amount of 48.57% apply the following steps of CDE. Nearly half of those (22.86%) use CD in an educational context. About the same amount (20.00%) use the term DevOps and one of the selected papers name DevSecOps (2.86%). [1], [2], [5], [31]–[62]

B. RQ2: In which courses and on which degree level (undergraduate or graduate) are CP taught to students, according to scientific literature?

Table IV summarises the courses and graduate levels named in the selected publications. It shows, that many of them use CP in Software Engineering courses (17 named at an undergraduate and 3 at graduate level). [2], [31]–[33], [35]–[38], [43], [44], [46]–[48], [51], [54], [57], [59]

There even are courses which specialise on DevOps (some of them with additional course contents like Cloud-Computing). 3 of them are taught at an undergraduate and 4 of them on a graduate level. [34], [39], [49], [53], [55], [56], [58]

3 courses use CP to teach students software development (programming). All of them use CI only. This practice is used to receive and analyse students code and to provide instant feedback to them. This continuous feedback improves code quality and increases the learning success of students. [41], [48], [60]

TABLE IV: Courses and graduation levels mentioned in the selected publications

Course	Undergraduate	Graduate
Software Engineering	**17**	**3**
DevOps (+ additional)	**3**	**4**
Software Development and Programming	**3**	**0**
Others	**3**	**0**
Software Architecture	**0**	**2**

TABLE V: Clustering of publications in different categories

Category	Subcategory	Percent
Active		**34.29%**
	DevOps and agile methods	31.43%
	Engineering	2.86%
	Development	0.00%
Passive		**60.00%**
	DevOps and agile methods	8.57%
	Engineering	2.86%
	Development	48.57%

Courses which only got named once, were summarized in the category "Others" (see Table IV). [48], [51], [52]

In Software Architecture courses two publications applied CP on a graduate level. [1], [57]

7 publications do not give any further information about the courses they are applying CP at. [5], [40], [42], [45], [50], [61], [62]

The sum of these values will not equal 35 (if the 7 papers without further information are added). This is due to the issue, that many publications mention multiple courses, in which they apply their approach.

C. RQ3: How are CP taught in higher computer science education, according to scientific literature?

First, the papers will be clustered in the categories *1) Active* (students do have to set up any kind of CP by themselves) and *2) Passive* (students use an existing setup of CP). Taking the different courses as a basis, additional categories of what they want to teach their students have been developed by the authors. Those categories were selected by the overall tasks the students had to perform, according to the publication. The categories are: *DevOps and agile methods, engineering, development*. They will both be found in the active and passive categories. Table V visualizes this clustering.

1) Active: This category contains 12 of 35 selected publications, where students have to actively setup CP and configure them to work together (equals 34.29%). Two of the papers ([2], [58]) describe concepts, which have not been actively applied so far. They will not be categorised in the following and also not be included in Table V.

DevOps and agile methods: Most of the publications, where students have to actively setup CP, can be categorized here (11 out of 35 equals 31.43%). They aim to teach students how these practices work (in real-world scenarios, within agile teams etc.). Therefore, a "hands-on approach" was chosen by all these papers. Students have to gain a deeper theoretical understanding to be able to setup a working toolchain. Also the cultural aspects of DevOps are taught, when students have to apply them in agile team scenarios. Often they will be confronted with real world problems (of technical nature or in team collaboration). [1], [34], [39], [40], [42], [49], [53], [55]–[57], [59]

Engineering: This category includes publications where, students have to actively focus on software engineering and use CP. Collaborations between industry and academy are, for example, used to teach CP and software engineering [32]. This helps students to gain a deeper understanding of when to use CP when working with real world customers. They have to manage requirements, develop and improve their development by continuous customer feedback. However, only this single publication was categorised into this section.

Development: In non of the selected publications, students have to setup CP to be taught in software development / programming.

2) Passive: This category contains 21 of 35 selected publications (equals 60.00%). Here, students do not have to setup CP by themselves, but use them.

DevOps and agile methods: Setting up CP can be a time consuming and difficult task for students. 3 of 35 (equals 8.57%) publications only focus on teaching CP and some apply already existing and running solutions. They will gain the same understanding of CP as the ones who actively setup those tools. But they will lack of technical skills and experience. Nevertheless, more theoretical and operational aspects of CP can be illuminated, as students do not spend time on installations. [33], [46], [61]

Engineering: One paper is categorized here. The authors describe an online software engineering course, where students are globally distributed and have to work together. To simplify their collaboration, CI tools are used. [47]

Development: 17 of 35 papers are located in this category (equals 48.57%), which makes it the largest. The publications in this category apply CP to teach software development (programming) to students. CP provide an opportunity to make feedback continuously available for students. Using these, bugs can be located, code quality be improved etc. Continuous assessment seems to be one of the biggest achievements of CP in education. [5], [31], [35]–[38], [41], [43]–[45], [48], [50]–[52], [54], [60], [62]

V. Conclusion

This SLR has discussed the application of CP in higher computer science education. Three research questions were identified and systematically answered. This has shown, that different kind of CP are used in various educational contexts. Many papers use CP in an active manner. A larger number of selected papers use them passively. However, all of them want to gain a deeper understanding for this difficult but important topic by students.

REFERENCES

[1] L. Greising, A. Bartel, and G. Hagel, "Introducing a deployment pipeline for continuous delivery in a software architecture course," in *Proceedings of the 3rd European Conference of Software Engineering Education*, ser. ECSEE'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 102107. Available: https://doi.org/10.1145/3209087.3209091

[2] B. P. Eddy, N. Wilde, N. A. Cooper, B. Mishra, V. S. Gamboa, K. N. Patel, and K. M. Shah, "Cdep: Continuous delivery educational pipeline," in *Proceedings of the SouthEast Conference*, ser. ACM SE '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 5562. Available: https://doi.org/10.1145/3077286.3077301

[3] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017. Available: https://ieeexplore.ieee.org/abstract/document/7884954

[4] E. Laukkanen, J. Itkonen, and C. Lassenius, "Problems, causes and solutions when adopting continuous deliverya systematic literature review," *Information and Software Technology*, vol. 82, pp. 55 – 79, 2017. Available: http://www.sciencedirect.com/science/article/pii/S0950584916302324

[5] G. Rong, S. Gu, H. Zhang, and D. Shao, "Devopsenvy: An education support system for devops," in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*, 2017, pp. 37–46.

[6] F. Erich, C. Amrit, and M. Daneva, "Report: Devops literature review," Tech. Rep., 10 2014.

[7] D. Ståhl, T. Martensson, and J. Bosch, "Continuous practices and devops: beyond the buzz, what does it all mean?" in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2017, pp. 440–448. Available: https://ieeexplore.ieee.org/document/8114695

[8] S. Badshah, A. A. Khan, and B. Khan, "Towards process improvement in devops: A systematic literature review," in *Proceedings of the Evaluation and Assessment in Software Engineering*, ser. EASE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 427433. Available: https://doi.org/10.1145/3383219.3383280

[9] F. Erich, C. Amrit, and M. Daneva, "Cooperation between information system development and operations: A literature review," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '14. New York, NY, USA: Association for Computing Machinery, 2014. Available: https://doi.org/10.1145/2652524.2652598

[10] T. Mårtensson, D. Ståhl, A. Martini, and J. Bosch, "Continuous architecture: Towards the goldilocks zone and away from vicious circles," in *2019 IEEE International Conference on Software Architecture (ICSA)*, 2019, pp. 131–140.

[11] M. Shahin, M. Zahedi, M. A. Babar, and L. Zhu, "Adopting continuous delivery and deployment: Impacts on team structures, collaboration and responsibilities," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 384393. Available: https://doi.org/10.1145/3084226.3084263

[12] H. Myrbakken and R. Colomo-Palacios, "Devsecops: A multivocal literature review," in *Software Process Improvement and Capability Determination*, A. Mas, A. Mesquida, R. V. O'Connor, T. Rout, and A. Dorling, Eds. Cham: Springer International Publishing, 2017, pp. 17–29.

[13] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, and A. Gumaei, "Prioritization based taxonomy of devops security challenges using promethee," *IEEE Access*, vol. 8, pp. 105 426–105 446, 2020.

[14] D. Ståhl and J. Bosch, "Modeling continuous integration practice differences in industry software development," *Journal of Systems and Software*, vol. 87, pp. 48 – 59, 2014. Available: http://www.sciencedirect.com/science/article/pii/S0164121213002276

[15] A. Eck, F. Uebernickel, and W. Brenner, "Fit for continuous integration: How organizations assimilate an agile practice," in *AMCIS*, 2014.

[16] P. Rodríguez, A. Haghighatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, and M. Oivo, "Continuous deployment of software intensive products and services: A systematic mapping study," *Journal of Systems and Software*, vol. 123, pp. 263 – 291, 2017. Available: http://www.sciencedirect.com/science/article/pii/S0164121215002812

[17] E. Laukkanen and M. Mntyl, "Build waiting time in continuous integration – an initial interdisciplinary literature review," in *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*, 2015, pp. 1–4.

[18] D. G. Widder, M. Hilton, C. Kästner, and B. Vasilescu, "A conceptual replication of continuous integration pain points in the context of travis ci," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 647658. Available: https://doi.org/10.1145/3338906.3338922

[19] M. A. Mascheroni and E. Irrazábal1, "Continuous Testing and Solutions for Testing Problems in Continuous Delivery: A Systematic Literature Review," *Computación y Sistemas*, vol. 22, pp. 1009 – 1038, 09 2018. Available: https://doi.org/10.13053/cys-22-3-2794

[20] T. E. Cardoso, A. R. Santos, R. Chanin, and A. Sales, "Change management practices for continuous delivery - a systematic literature mapping," in *Software Business*, S. Hyrynsalmi, M. Suoranta, A. Nguyen-Duc, P. Tyrväinen, and P. Abrahamsson, Eds. Cham: Springer International Publishing, 2019, pp. 175–182.

[21] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A survey of devops concepts and challenges," *ACM Comput. Surv.*, vol. 52, no. 6, Nov. 2019. Available: https://doi.org/10.1145/3359981

[22] S. Rafi, W. Yu, and M. A. Akbar, "Rmdevops: A road map for improvement in devops activities in context of software organizations," in *Proceedings of the Evaluation and Assessment in Software Engineering*, ser. EASE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 413418. Available: https://doi.org/10.1145/3383219.3383278

[23] S. G. Yaman, T. Sauvola, L. Riungu-Kalliosaari, L. Hokkanen, P. Kuvaja, M. Oivo, and T. Männistö, "Customer involvement in continuous deployment: A systematic literature review," in *Requirements Engineering: Foundation for Software Quality*, M. Daneva and O. Pastor, Eds. Cham: Springer International Publishing, 2016, pp. 249–265.

[24] M. Fernandes, S. Ferino, U. Kulesza, and E. Aranha, "Challenges and recommendations in devops education: A systematic literature review," in *Proceedings of the 34th Brazilian Symposium on Software Engineering*, ser. SBES '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 648657. Available: https://doi.org/10.1145/3422392.3422496

[25] B. Kitchenham, "Procedure for undertaking systematic reviews," Computer Science Department, Keele University (TRISE-0401) and National ICT Australia Ltd (0400011T. 1), Tech. Rep., 2004, joint Technical Report. Available: http://www.it.hiof.no/~haraldh/misc/2016-08-22-smat/Kitchenham-Systematic-Review-2004.pdf

[26] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE Technical Report EBSE-2007-01, Keele University and University of Durham, Tech. Rep., 2007. Available: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[27] L. Chen, M. A. Babar, and H. Zhang, "Towards an evidence-based understanding of electronic data sources," in *14th International Conference on Evaluation and Assessment in Software Engineering (EASE) (EASE)*, 2010. Available: https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/EASE2010.17

[28] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571 – 583, 2007, software Performance. Available: http://www.sciencedirect.com/science/article/pii/S016412120600197X

[29] G. Booch, *Object-oriented Analysis and Design with Applications*, ser. Benjamin/Cummings series in object-oriented software engineering. Benjamin/Cummings Publishing Company, 1994. Available: https://books.google.de/books?id=9mUPAQAAMAAJ

[30] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, "Using mapping studies in software engineering," *Proceedings of PPIG 2008*, vol. 2, 01 2008.

[31] B. P. Eddy, N. Wilde, N. A. Cooper, B. Mishra, V. S. Gamboa, K. M. Shah, A. M. Deleon, and N. A. Shields, "A pilot study on introducing continuous integration and delivery into undergraduate software engineering courses," in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*, 2017, pp. 47–56.

[32] K. Kuusinen and S. Albertsen, "Industry-academy collaboration in teaching devops and continuous delivery to software engineering students:

Towards improved industrial relevance in higher education," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2019, pp. 23–27.

[33] X. Bai, M. Li, D. Pei, S. Li, and D. Ye, "Continuous delivery of personalized assessment and feedback in agile software engineering projects," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2018, pp. 58–67.

[34] R. A. K. Jennings and G. Gannod, "Devops - preparing students for professional practice," in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–5.

[35] X. Bai, D. Pei, M. Li, and S. Li, "The devops lab platform for managing diversified projects in educating agile software engineering," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5.

[36] Z. Hu and E. F. Gehringer, "Improving feedback on github pull requests: A bots approach," in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–9.

[37] J. G. Süß and W. Billingsley, "Using continuous integration of code and content to teach software engineering with limited resources," in *2012 34th International Conference on Software Engineering (ICSE)*, 2012, pp. 1175–1184.

[38] K. Presler-Marshall, E. Horton, S. Heckman, and K. Stolee, "Wait, wait. no, tell me. analyzing selenium configuration effects on test flakiness," in *2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST)*, 2019, pp. 7–13.

[39] Y. Demchenko, Z. Zhao, J. Surbiryala, S. Koulouzis, Z. Shi, X. Liao, and J. Gordiyenko, "Teaching devops and cloud based software engineering in university curricula," in *2019 15th International Conference on eScience (eScience)*, 2019, pp. 548–552.

[40] Z. Ahmed and S. C. Francis, "Integrating security with devsecops: Techniques and challenges," in *2019 International Conference on Digitization (ICD)*, 2019, pp. 178–182.

[41] C. Mendoza, K. Garcs, R. Casallas, and J. Bocanegra, "Detecting architectural issues during the continuous integration pipeline," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 589–597.

[42] S. Krusche and L. Alperowitz, "Introduction of continuous delivery in multi-customer project courses," in *Companion Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE Companion 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 335343. Available: https://doi.org/10.1145/2591062.2591163

[43] J. Bowyer and J. Hughes, "Assessing undergraduate experience of continuous integration and test-driven development," in *Proceedings of the 28th International Conference on Software Engineering*, ser. ICSE '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 691694. Available: https://doi.org/10.1145/1134285.1134393

[44] W. Billingsley and J. Steel, "A comparison of two iterations of a software studio course based on continuous integration," in *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 213218. Available: https://doi.org/10.1145/2462476.2465592

[45] M. Ohtsuki, K. Ohta, and T. Kakeshita, "Software engineer education support system alecss utilizing devops tools," in *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*, ser. iiWAS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 209213. Available: https://doi.org/10.1145/3011141.3011200

[46] B. T. Bennett and M. L. Barrett, "Incorporating devops into undergraduate software engineering courses: A suggested framework," *J. Comput. Sci. Coll.*, vol. 34, no. 2, p. 180187, Dec. 2018.

[47] W. Billingsley, R. Torbay, P. R. Fletcher, R. N. Thomas, J. R. H. Steel, and J. G. Süß, "Taking a studio course in distributed software engineering from a large local cohort to a small global cohort," *ACM Trans. Comput. Educ.*, vol. 19, no. 2, Jan. 2019. Available: https://doi.org/10.1145/3218284

[48] S. Heckman and J. King, "Developing software engineering skills using real tools for automated grading," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 794799. Available: https://doi.org/10.1145/3159450.3159595

[49] H. B. Christensen, "Teaching devops and cloud computing using a cognitive apprenticeship and story-telling approach," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 174179. Available: https://doi.org/10.1145/2899415.2899426

[50] C. Geigle, I. Lourentzou, H. Sundaram, and C. Zhai, "Clads: A cloud-based virtual lab for the delivery of scalable hands-on assignments for practical data science education," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 176181. Available: https://doi.org/10.1145/3197091.3197135

[51] S. Krusche and A. Seitz, "Artemis: An automatic assessment management system for interactive learning," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 284289. Available: https://doi.org/10.1145/3159450.3159602

[52] P. Gestwicki, "The entity system architecture and its application in an undergraduate game development studio," in *Proceedings of the International Conference on the Foundations of Digital Games*, ser. FDG '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 7380. Available: https://doi.org/10.1145/2282338.2282356

[53] M. Airaj, "Enable cloud devops approach for industry and higher education," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 5, p. e3937, 2017, e3937 CPE-15-0382.R1. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3937

[54] S. M. Embury and C. Page, "Effect of continuous integration on build health in undergraduate team projects," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, J.-M. Bruel, M. Mazzara, and B. Meyer, Eds. Cham: Springer International Publishing, 2019, pp. 169–183.

[55] A. Capozucca, N. Guelfi, and B. Ries, "Design of a (yet another?) devops course," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, J.-M. Bruel, M. Mazzara, and B. Meyer, Eds. Cham: Springer International Publishing, 2019, pp. 1–18.

[56] J. Kousa, P. Ihantola, A. Hellas, and M. Luukkainen, "Teaching container-based devops practices," in *Web Engineering*, M. Bielikova, T. Mikkonen, and C. Pautasso, Eds. Cham: Springer International Publishing, 2020, pp. 494–502.

[57] B. Benni, P. Collet, G. Molines, S. Mosser, and A.-M. Pinna-Déry, "Teaching devops at the graduate level," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, J.-M. Bruel, M. Mazzara, and B. Meyer, Eds. Cham: Springer International Publishing, 2019, pp. 60–72.

[58] A. Capozucca and N. Guelfi, "Analysing the swecom standard for designing a devops education programme," in *Frontiers in Software Engineering Education*, J.-M. Bruel, A. Capozucca, M. Mazzara, B. Meyer, A. Naumchev, and A. Sadovykh, Eds. Cham: Springer International Publishing, 2020, pp. 133–150.

[59] E. Bobrov, A. Bucchiarone, A. Capozucca, N. Guelfi, M. Mazzara, and S. Masyagin, "Teaching devops in academia and industry: Reflections and vision," in *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, J.-M. Bruel, M. Mazzara, and B. Meyer, Eds. Cham: Springer International Publishing, 2020, pp. 1–14.

[60] Y.-Z. Cai and M.-H. Tsai, "Improving programming education quality with automatic grading system," in *Innovative Technologies and Learning*, L. Rønningsbakk, T.-T. Wu, F. E. Sandnes, and Y.-M. Huang, Eds. Cham: Springer International Publishing, 2019, pp. 207–215.

[61] B. Bruegge, S. Krusche, and M. Wagner, "Teaching tornado: From communication models to releases," in *Proceedings of the 8th Edition of the Educators' Symposium*, ser. EduSymp '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 512. Available: https://doi.org/10.1145/2425936.2425938

[62] C. Matthies, A. Treffer, and M. Uflacker, "Prof. ci: Employing continuous integration services and github workflows to teach test-driven development," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–8.