Understanding DevOps Education with Grounded Theory

Candy Pang
Computer Science
MacEwan University
Edmonton, Alberta, Canada
candy.pang@macewan.ca

Abram Hindle
Computing Science
University of Alberta
Edmonton, Alberta, Canada
abram.hindle@ualberta.ca

Denilson Barbosa
Computing Science
University of Alberta
Edmonton, Alberta, Canada
denilson@ualberta.ca

ABSTRACT

DevOps stands for Development-Operations. It arises from the IT industry as a movement aligning development and operations teams. DevOps is broadly recognized as an IT standard, and there is high demand for DevOps practitioners in industry. Since ACM & IEEE suggest that undergraduate computer science curricula "must adequately prepare [students] for the workforce", we studied whether undergraduates acquired adequate DevOps skills to fulfill the demand for DevOps practitioners in industry. We employed Grounded Theory (GT), a social science qualitative research methodology, to study DevOps education from academic and industrial perspectives. In academia, academics were not motivated to learn or adopt DevOps, and we did not find strong evidence of academics teaching DevOps. Academics need incentives to adopt DevOps, in order to stimulate interest in teaching DevOps. In industry, DevOps practitioners lack clearly defined roles and responsibilities, for the DevOps topic is diverse and growing too fast. Therefore, practitioners can only learn DevOps through hands-on working experience. As a result, academic institutions should provide fundamental DevOps education (in culture, procedure, and technology) to prepare students for their future DevOps advancement in industry. Based on our findings, we proposed five groups of future studies to advance DevOps education in academia.

KEYWORDS

Software Engineering, Continuous Integration, Continuous Delivery, DevOps, Grounded Theory, Education

1 Introduction

DevOps stands for Development-Operations, is an important software engineering topic arising from the IT industry in 2009. It is a movement aligning development and operations teams in "organizational structure, practices, and culture to enable rapid development and scalable, reliable operations" [1]. According to Google Search, DevOps is one of the most searched terms in technology over the last five years, and continues to rise [2]. According to Gartner in 2018, "DevOps processes would drive over 80% of the industrial-strength technology on the market" [3]. Currently, there is a high demand for DevOps practitioners, with many related job postings. DevOps is on its way becoming a required skill for IT practitioners. According to ACM & IEEE Guidelines for Undergraduate Degree Programs in Computer Science, "the education that undergraduates in computer science receive must adequately prepare them for the workforce" [4]. Therefore, we studied the current state of DevOps education, and how DevOps education should proceed in the future.

Since DevOps is very popular in industry, there are many blogs, webinars, conferences, communities, and organizations in industry trying to understand, promote, assist, and improve DevOps. However, de Franca *et al.* found that there is limited academic research on DevOps [5] [6]. To the best of our knowledge, no research has studied the state of DevOps education.

DevOps education is different from other kind of technical training, for DevOps is much more than a specific technology. Many IT practitioners argue that DevOps is a cultural, procedural, and technological movement [1] [7] [8] [9] [10] [11] [12] [13]. Therefore, we employed Grounded Theory (GT), a social science qualitative research methodology, to study DevOps education. GT allowed us to discover the state of DevOps education from external data without making presumptions.

Following the GT process, we started with interviewing IT practitioners. Then, through GT's open coding and microanalysis, we discovered that IT practitioners were interested in DevOps education and training. Continuing with the GT process, we used data analysis, interviews, and surveys to examine the current state and challenges of DevOps education from academic and industrial perspectives. Based on our findings, we propose future research direction for DevOps education.

From the academic perspective, we found that most institutions were not teaching their undergraduate computer science students much about DevOps. Academics were not motivated to learn or adopt DevOps, for they thought the cost of adopting DevOps could not be justified in their research projects. There needs to be more incentives for academics to adopt DevOps, in order to stimulate interest in teaching DevOps.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ICSE-SEET'20, May 23–29, 2020, Seoul, Republic of Korea
© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7124-7/20/05…$15.00
https://doi.org/10.1145/3377814.3381711

From the industrial perspective, we found that industry has not clearly defined the roles and responsibilities for DevOps practitioners. Therefore, it is not clear what students should learn to become DevOps practitioners. In addition, the DevOps community was evolving too fast for a single DevOps curriculum to be defined. Finally, DevOps experts asserted that classroom education was only the starting point in learning DevOps. Practitioners advance their DevOps skills at work. In conclusion, academic institutions should provide fundamental DevOps education (in culture, procedure, and technology) to prepare students for their future DevOps advancement in industry.

From our findings, we proposed five groups of hypotheses to study how to integrate DevOps culture, procedure, and technology into computer science education.

2 Background

To fulfill the great demand for DevOps practitioners, we want to find out how institutions can teach DevOps. We use grounded theory (GT) to study the current state of DevOps education. In this section, we present a brief DevOps history, and introduce the GT methodology.

2.1 Brief DevOps History

In the early days of software development, module integration usually occurred near the end of a project and often failed, which caused much frustration [14]. To address the issue, Continuous Integration (CI) was proposed as a practice that executes module integration frequently throughout a project lifecycle in order to avoid tension at the end of the project.

Most IT practitioners credited the CI concept to Extreme Programming (XP) [15]. In XP, Beck suggested that "developers should be integrating and committing code into the code repository every few hours, whenever possible". However, Beck was only proposing a CI concept, without any action plan.

In 2007, Duvall *et al.* wrote the first CI book [14] that detailed six CI tasks. In 2010, Humble *et al.* expanded the CI concept into Continuous Delivery (CD) [16]. Humble *et al.* described the CD process as a Deployment Pipeline. Then, DevOps expands CI and CD, from development to operations, by combining Development (Dev) and Operations (Ops). The name DevOps originates from the presentation "10+ Deploys per Day: Dev and Ops Cooperation at Flikr" by John Allspaw and Paul Hammond in 2009 at the O'Reilly's Velocity conference [17].

Technically, CI, CD, and DevOps are not the same. However, most people in the IT industry presume DevOps covers CI and CD. Therefore, we will use the term "DevOps" as an umbrella term to cover all continuous processes in the rest of this paper. Throughout the project, we used the terms CI, CD, and DevOps in different surveys, interviews, and documents at the time.

2.2 Grounded Theory Methodology

Grounded theory (GT) is a qualitative research methodology. GT uses "nonmathematical process of interpretation … to discover concepts and relationships in raw data and then organizes these results into a theoretical explanatory scheme" [18]. In 1967, Glaser and Strauss first described grounded theory in their seminal book [19]. "The goal of GT is to generate theory rather than test or validate existing theory" [20]. Since 1967, many variants of GT have been published. We analyzed and selected to follow Strauss and Corbin version of GT (S&C GT) [18].

The S&C GT suggested many data analyzing techniques, such as coding (open coding, axial coding, and selective coding), theoretical sampling, memos, conditional/consequential matrix, and diagrams. S&C GT uses coding to conceptualize, reduce, elaborate, and relate concepts. They, then use non-statistical sampling, writing of memos, and diagramming to analyze the information. Not all GT techniques are applicable to this project. The techniques we employed are defined below.

Open coding: a process where investigators go through the raw data line-by-line to identify initial concepts, properties, and dimensions from the data.

Microanalysis: a process that examines the initial concepts, properties, and dimensions from open coding to generate initial categories, and studies the relationships among these categories.

Selective coding: a process of integrating and refining a theory.

Theoretical saturation: a point in theory development at which no new properties, dimensions, or relationships emerge during analysis.

The S&C GT includes two phases. In the first phase, researchers gather raw data from the stakeholders, then microanalyze the raw data through open coding, and write memos to record ideas, concepts, thoughts, and deliberations during the analysis. At the end, researchers organize these concepts into discrete categories according to their properties and dimensions; then select one of the categories for further study. For the selected categories, the researchers define a potential theory to be studied in the second phase.

In the second phase, the researchers formulated the potential theory into logical, systematic, and explanatory dimensions. The dimensions are further developed and analyzed from many different angles through selective coding, until the theory is theoretically saturated and finalized.

2.3 Related Work

Even through GT is a social sciences methodology, there is existing DevOps research using GT, but not on DevOps education. Franca *et al.* used a GT inspired procedure to analyze a broad range of academic literature to define DevOps [5], but the DevOps definition and scope remains unclear. Similarly, Erich *et al.* used a GT inspired approach to study DevOps publications, and found that the quantity and quality of DevOps academic studies is low [6]. Erich *et al.* turned to industry to further their DevOps study. Adolph *et al.* used GT to study how individual aptitudes and team social factors affect software development cost [21]. Hilton *et al.* used GT processes to study developers' barriers and needs when using CI [22]. The study concluded that developers had to make trade-off decisions between speed and stability; accessibility and security; flexibility and simplicity when using CI. Hilton *et al.* has also used a qualitative methodology to study the costs, benefits, and usage of CI in open-source software [23]. The authors concluded from their GT project that "developers encounter[ed] increased complexity, increased time costs, and new security concerns when working with CI" [23].

Stol *et al.* compared various GT variants, and studied 98 software engineering publications that claimed to use GT [20]. However, Stol *et al.* found that most of these publications were "slurring" GT. Based on the suggestions by Stol *et al.*, we decided to create a compliant GT study on DevOps.

3 Open-Ended DevOps Study

Following the S&C GT [18], we started a DevOps study. S&C GT suggested that investigators of a real GT project should start with open minds, and without any pre-determined agenda. We paid attention in not allowing our knowledge and preferences to affect the GT project.

In the first phase of the GT project, we interviewed four IT practitioners regarding their DevOps experiences with repositories, build scripts, integrated development environments (IDEs), testing, code quality, and deployment. In interview order, the interviewees had 6, 10, 30, and 15 years of IT experience, with IT skills ranging from intermediate to advanced. The interviewees provided IT services in different fields, with different responsibilities, but they had all participated in software development. We transcribed the interviews, and applied open coding to the transcriptions. The coding results showed that the volunteers have consistent and similar ideas about DevOps. GT relies on quality, rather than quantity. We found this diverse group of volunteers sufficiently covered enough of the DevOps spectrum. Therefore, we sought no further participants.

The first two authors of this paper followed the S&T GT open coding process to microanalyze the transcriptions individually. Then, we got together to compare codes, line by line. We shared, compared, communicated, discussed, and enhanced the codes to agree on a final list of code. We finalized 34, 83, 28, and 125 codes from the four transcriptions respectively. There were 153 unique codes. Three of the 153 unique codes appeared most often: "experience" appeared 12 times, "automation" appeared 8 times, and "reduce defect" appeared 7 times. In addition, 90 codes appeared only once, 36 codes appeared twice, 17 codes appeared 3 times, 5 codes appeared 4 times, and 2 codes appeared 5 times. The codes covered a broad range of concepts across the DevOps spectrum.

Then, we manually analyzed the codes, and linked related codes into seven categories, named "experience", "practice", "benefit", "cost", "tool", "social aspect", and "third-party service". These categories have 19, 50, 28, 14, 30, 18, and 6 unique codes respectively. Throughout the coding and analysis processes, we recorded thoughts, ideas, concepts, and questions in memos. After the code analysis, we reviewed the memos and identified 119 potential research questions. From these categories and potential research questions, we needed to select one topic for in-depth GT investigation in the second phase. The selected topic should have the potential to provide the highest value to stakeholders.

Among all the codes only "cost", "reduce defect", and "training" appeared in all four interviews. These three DevOps codes had surely drawn attention from IT practitioners. In addition, there were 16 other codes that appeared in three out of four interviews. The categories "experience", "practice", "benefit", "cost", and "tool" have 3, 7, 5, 4, and 2 codes that appeared in three or more interviews respectively. The results showed that these categories drew interest among DevOps practitioners.

As other research has identified, DevOps lacks common definition and scope [5]. In depth study of DevOps "practice", "benefit", "cost", and "tool" relies on common DevOps definitions. Therefore, "experience" remained as a potential category. In the "experience" category, the code "training" appeared in all four of the interviews. Practitioners need appropriate training to learn DevOps. As one of the interviewees said, "Whatever tools you know actually guide your practice and how you will apply DevOps." Training strongly influenced what DevOps practices were adopted. As the result, we decided to study DevOps education further in the second phase.

After the topic was selected, we reviewed the transcriptions, memos, code analysis, and research questions to define a potential theory regarding DevOps education, which is "Continuous Integration (CI) is a craftsmanship. How do you learn the craft?" In the second phase, we followed the GT process to enforce, enhance, refine, reject, or redefine this theory.

4 DevOps Education Findings

In the first phase, we selected DevOps education as the topic and defined a potential theory. In the second phase, we examined this potential theory from academic and industrial perspectives, through questioning, selective coding, surveying, data gathering, and making comparisons. At the end, a DevOps education theory emerged from the analysis.

4.1 DevOps in Academic Education

To study whether academic education delivered DevOps knowledge, we analyzed the computer science curricula of the top 50 institutions according to the 2017 QS World University Rankings by Subject - Engineering and Technology [24]. The data showed that computer science curricula covered limited DevOps knowledge.

4.1.1 Search Process Combining the six CI tasks defined by Duvall *et al.* [14] and the open coding results in phase one, we identified five groups of significant DevOps terms as listed below with variations in parenthesis.

1. Continuous Integration (continuous, integration, delivery)
2. Testing (testing, test)
3. Build (auto, build)
4. Repository (repository, version control, git, cvs)
5. Deployment (deploy, deployment, release)

We searched these DevOps terms in the curricula, and reviewed the context to make sure the curricula actually included DevOps related content.

4.1.2 Search Result We excluded 11 institutions from the 50 because they have no English course descriptions, or their course descriptions were only accessible by authenticated users. Some institutions provided course outlines only, other provided detailed descriptions or curriculum. We always explored the most detailed description available. Table 1 shows the search result.

4.1.3 Result Analysis From the search result, only 5 out of 39 (12.8%) institutions included CI; 5 out of 39 (12.8%) included explicitly build processes; 11 out of 39 (28.2%) included repositories; 9 out of 39 (23.1%) included deployment. Nevertheless, 35 out of 39 (89.7%) institutions included software testing. Hence, DevOps and its corresponding tasks were not part of the curriculum in most institutions, except software testing.

Table 1 Number of courses with DevOps content

Institution	CI	Test	Build	Repo	Deploy
1	0	4	0	0	0
2	0	0	0	0	1
3	1	3	0	1	0
4	0	2	0	0	1
5	0	2	0	0	0
6	2	4	1	0	1
7	0	10	1	2	1
8	0	1	0	0	0
9	0	1	0	0	0
12	0	1	0	0	0
13	0	2	0	0	0
14	0	1	1	0	0
15	0	3	0	0	0
16	0	3	1	1	0
17	0	4	0	1	0
=18	0	0	0	0	0
20	0	2	0	0	0
21	0	0	0	0	0
=24	0	1	0	0	0
27	0	2	0	0	0
28	0	3	0	0	0
29	1	5	1	1	0
30	0	3	0	0	0
=31	0	0	0	0	0
=31	0	7	0	0	0
34	0	3	0	1	1
=35	0	5	0	0	0
37	0	1	0	0	0
=38	0	1	0	0	0
=38	0	4	0	0	0
41	0	5	0	1	1
42	0	1	0	0	0
43	1	8	0	2	1
=44	3	8	0	2	2
=44	0	2	0	0	0
46	0	4	0	1	1
=47	0	4	0	1	0
49	0	3	0	0	0
50	0	3	0	0	0

4.1.3.1 Software Testing Teaching Almost all institutions taught software testing. One possible explanation is that software testing has always been an important component of software engineering. Myers published the book "The Art of Software Testing" [25] in 1979. Myers found that "students graduate and move into industry without any substantial knowledge of how to go about testing a program" [25], therefore he wrote the book to satisfy the need. In that case, DevOps may be added to the computer science curriculum in the future, as DevOps is becoming essential to students going into industry.

Software testing continues to be an important component of software engineering. In 1999, Back introduced Test-Driven Development (TDD) [15]. TDD emphasizes that developers should program test cases before programming functionalities. In 2007, Duvall *et al.* [14] defined test automation as a DevOps requirement. However, by most of the course description, we could not tell whether the courses included TDD or automated testing. If we looked strictly for automated software testing in the curriculum, then the number of courses would be much lower.

We also realized that most of the courses identified in Table 1 were advanced software engineering courses, but none were dedicated to DevOps. Hence, DevOps was only a topic in a larger software engineering course. For testing, six institutions (#14, #17, #20, #35, #38, #44) have courses dedicated to automated testing. Two institutions (#41, #43) have dedicated courses about version control and testing. Academic education has embraced testing, one of the DevOps topics, but not yet DevOps.

4.1.3.2 Challenges in DevOps Teaching During the search, we also realized that it might not be easy to teach some of the DevOps skills in an academic setting. For example, to let students practice automated software build and deployment, students needed designated build and deployment infrastructure. It might be a challenge to configure such infrastructure for many students in an academic environment.

Nonetheless, the course "COMP1040 The Craft of Computing" offered by institution #46 agreed with the potential theory that software development was craftsmanship, a form of art. "The focus of The Craft of Computing [is] on understanding core principles that allow students to quickly and confidently learn and apply a variety of computational tools to several different types of problems" [26]. The course description also indicated that academic education could only equip students with the fundamental knowledge. Students needed to advance their comprehension and understanding through hands-on experience.

Academic education also had difficulty catching up with the fast-growing industrial practice. As the Department of Computer Science at institution #45 indicated, "The [Computer Science] faculty revised the curriculum, because computer science has become too large to learn in four years" [27]. Computer science research progresses and day-to-day applications are expanding rapidly in academia and everyday lives. Institutions are challenged to define curricula that would properly equip students to be IT practitioners. To provide comprehensive DevOps education, institutions need to add much more content to the already packed four-year curriculum. Therefore, academic education can cover, at most, high-level DevOps knowledge.

With the ever-expanding amount of computer science knowledge, some institutions chose to focus more on DevOps. From Table 1, institutions offered on average 3.92 DevOps related courses and the median was 3 courses. However, institution #7 offered 14; #43 offered 12; and #44 offered 15. On the other hand, many more institutions did not address DevOps. Three institutes (#18, #21, #31) did not offer any DevOps related courses. Institution #2 has only one. Seven other institutions offered only one course about testing. Different institutions have different favourite areas of teaching.

In conclusion, DevOps is being covered as a secondary topic and inconsistently across top institutions. With DevOps becoming an industrial standard, all institutions should start conveying basic DevOps knowledge, such as the six fundamental CI tasks defined by Duvall *et al.* [14].

4.2 DevOps in Academic Research

If academics are not interested in DevOps, it is unlikely that they have the knowledge or interest to teach students about DevOps. In this section, we examine the popularity of DevOps among academics.

4.2.1 Research Process Graduate students usually implement computer science research projects, under the supervision of computer science professors. We invited all graduate students in our local computer science department for interviews to learn whether they practiced DevOps in their research projects. In addition, we invited research groups from different research areas for interviews. We prepared the following interview questions.

1. Does your research involve any coding?
2. Do you think adopting CI in your situation will improve the quality of your research?
3. Do you think you would adopt CI in your situation? Why and why not?
4. What is the biggest barrier in adopting CI? Emotional? Intellectual? Technical?

After the interviews, we transcribed the interviews and applied selective coding. Through selective coding, we went through the transcriptions looking for data to verify or reject the potential theory. The potential theory was refined through the process.

4.2.2 Research Result After a month-long effort, all research groups turned down the invitation. As one of the groups responded, "nobody [in the group] seem[ed] to be interested in learning more about [DevOps]". The rejection was itself very valuable data for the GT project. It was clear that academics were not interested in DevOps. In the end, we interviewed three graduate students in different research areas. All three interviewees provided consistent answers to the questions. Therefore, we determined that there was sufficient data to saturate the theory and seek no addition interviewees.

For the first question, all interviewees developed software for their research projects. For the second question, all interviewees agreed that DevOps would improve the quality of their software projects. For the third and fourth questions, all interviewees were absolutely sure that they would NOT adopt CI in their research projects. They provided the following reasons.

1. Lacked knowledge and experience about DevOps.
2. DevOps sounded really complicated.
3. The DevOps learning curve was too steep.
4. Needed too much time and trial-and-error to set up DevOps.
5. DevOps tools were user unfriendly and lacked manual.
6. Lacked sufficient DevOps resources and support.
7. Research projects were too small and short for DevOps.
8. The overhead of setting up DevOps did not fit the tight timeline of research projects.
9. DevOps did not add enough value to research projects.
10. DevOps was only for commercial software projects.

A survey of 497 IT practitioners found similar barriers in industry [28]. In both academia and industry, the DevOps adoption concerns were lacking knowledge, lacking skills and resources, difficulties in setting up the environment, and constrained timelines.

4.2.3 Result Analysis The rejections and lack of volunteers provided evidence for low interest among academics. When we approached the graduate students, who had not worked in the IT industry, they either had never heard about DevOps, or had no interest in learning anything about it. The three volunteer interviewees had industrial working experience in different fields. Hence, DevOps drew interest from people with hands-on working experience only, as we had claimed in the potential theory. This phenomenon may hinder DevOps education, since undergraduate students are not likely to have IT working experience.

It was interesting that all three interviewees confirmed that they would not adopt DevOps in their research projects, even though DevOps would benefit their projects. They believed the cost of adopting DevOps was bigger than the benefit. In deeper discussion with the interviewees, replicability was the biggest benefit of DevOps. Replicability allowed research results to be verified by the public, which improved credibility of the result. However, higher replicability was not enough to persuade the interviewees to adopt DevOps. As summarized by one of the interviewees, "with [DevOps], I can make research result repeatable, but until it is commonly adopted by the whole research field, no one wants to spend the time to try to make it happen". If academia gives more credit to verifiable results, then there may be more interest in DevOps.

Since academics are not motivated to learn or practice DevOps, it is not likely that they have the knowledge or interest to teach DevOps. Much work needs to be done to improve DevOps recognition in academia.

4.3 DevOps Requirements in Industry

Both academics and industrial practitioners believed that lacking of DevOps knowledge and skills was a barrier. Therefore, we tried to identify DevOps knowledge and skills. Knowing what the required DevOps skills are, will clear the path for DevOps education.

4.3.1 DevOps Jobs Requirements DevOps rises from industry, so its requirements can only be found in industry. Lee searched through DevOps engineering job postings and identified eight groups of required skillsets corresponding to the eight groups of tasks, which we summarize as [29]:

1. Design, build, and operate technology stack.
2. Configure, monitor, and manage systems.
3. Program different tools in various languages.
4. Script Linux/Unix processes with various languages.
5. Operate different cloud products.
6. Automate IT processes regardless of tools.
7. Understand and enforce best practices.
8. Being proficient in interpersonal communication.

The requirements cover almost the whole spectrum of IT processes and tools. DevOps practitioners need to have the knowledge and skills of architect, programmer, scripter, system administrator, database administrator, cloud administrator, operational analyst, and more. They also need soft skills, including management, leadership, and communication. In conclusion, there are no clearly defined requirements for DevOps practitioners. The more experience the better. Lee suggested that DevOps practitioners should have at least five years of IT practitioners experience [30]. Hence, IT practitioners cannot become DevOps practitioners simply through education, without hands-on working experience. However, IT practitioners could learn fundamental DevOps knowledge from academic education, and advance their DevOps skills at work.

Delanbanque, a DevOps practitioner recruiting specialist, also agreed with Lee that DevOps job postings "change[d] entirely from company to company" [31]. Delanbanque suggested hiring "someone who under[stood] the methodologies over the tools" [31] for DevOps positions. As the potential theory suggested, DevOps is a form of art. Craftsman use different tools to create art pieces they comprehend through hands-on experience. "The focus should be on the candidates' ability to solve problems" [31]. Therefore, hands-on working experience is compulsory for DevOps practitioners.

4.3.2 Research Process and Result According to Lee [29], DevOps practitioners' requirements are broad and extensive, so we searched three types of job sites to see whether DevOps positions actually exist: a general public job site (Monster.com [32]), an IT practitioner specific job site (DZone.com [33]), and an academic specific job site (IEEE Computer Society Jobs Board [34]). Searching these job sites, we confirmed that DevOps jobs were abundant and popular.

Monster's postings are very broad and the number of jobs is very large, which increased the difficulty of the search. Monster posts its popular job titles in https://www.monster.com/jobs/job-title. Through analysis we decided to search for these titles: Application Integration Architect, Application Integration Engineer, Director Systems Integration, Software Integration Engineer, System Integration Analyst. As comparison, we also searched three commonly used software developer titles: Software Analyst, Software Architect, and Software Engineer. Since the number of postings in Monster was large, we had to use extra search criteria to limit the search. On December 13, 2017, we searched for new postings with those titles in the last 7 days in California. Table 2 shows the search results.

DZone is a community for software developers. Its members are experienced IT practitioners. Therefore, its postings focus on IT. DZone classifies job postings by categories. Employers can select any number of categories for their postings. On December 13, 2017, we searched DZone for postings using the categories "DevOps" and "Integration". On that date, DZone had about 100 postings. The "DevOps" category had 15 postings; seven with "DevOps" in the job titles, eight without. The "Integration" category had seven postings; one with "Integration" in the job titles, six without.

IEEE Computer Societys main audience is academic researchers and IT professionals. Therefore, its postings target academics, graduate students, and IT professionals. We searched IEEE Computer Society Jobs Board on December 14, 2017 for job postings with titles or description including the keywords "DevOps", "Continuous", and "Integration". On that date, the jobs board had 336 postings. Searching by the keyword "DevOps" returned one posting; the keyword "Continuous" returned 90 postings; the keyword "Integration" returned 62 postings.

4.3.3 Result Analysis Based on the search results, DevOps has definitely become a term used in job titles. We found DevOps as part of job titles in all three job sites.

On Monster, there were many more postings for Software Analyst (969) than Software Architect (173). Hence, the demand for Software Analysts is much higher than the demand for Software Architects. The number of postings for each title represents the corresponding demand in industry. Similarly, the number of postings for DevOps Engineer (489) was about half of those for Software Analyst (969) and almost three times of those for Software Architect (173). This represents a significant demand for DevOps practitioners. In addition, in DZone, 15 out of 100 postings were categorized as DevOps. Hence, DevOps represents a significant portion of IT positions among the IT practitioners, especially in the software development community.

Even though there were many postings for DevOps Engineer in Monster, sampling the job descriptions showed that most of the postings had either a development focus or an operations focus. Delanbanque had the same findings that employers were "looking for a candidate from either operations and infrastructure background or someone from a software engineering and development background" [31]. This condition indicates that the DevOps movement is still far from its original goal to integrate

Table 2 Monster's job titles and number of postings for the last 7 days in California on 2017-12-13

Job Titles	# Post
DevOps Engineer	489
Application Integration Engineer	1000+
Director Systems Integration	53
Software Integration Engineer	516
System Integration Analyst	35
System Integration Architect	17
System Integration Consultant	1000+
Software Analyst	969
Software Architect	173
Software Engineer	1000+

development with operations. However, to fulfill the original DevOps goal, IT practitioners need to have extensive knowledge and skills, as identified by Lee [29] in Section 4.3.1. It is not likely an academic education can cover the required extensive knowledge and skills. IT practitioners need to earn and enhance their DevOps knowledge and skills through hands-on experience.

Since DZone is a software developer community, the job requirements for DevOps practitioners in DZone's postings are much closer to those identified by Lee [29]. Only experienced IT practitioners understand what DevOps really stands for, and what kind of work experience is significant for DevOps. Human resource representatives without IT experience may not be able to specify accurate DevOps skills and knowledge requirements. If the IT industry does not know what skills they want to employ, then it is hard to design DevOps education. As the executive director at Robert Half Technology said, "At some point in the future you will be able to hire for just DevOps, [but] we're almost ahead of ourselves today. What you really need or want is still somewhat undefined." [35]

Finally, in Monster, many jobs with "continuous" and "integration" in their titles had nothing to do with DevOps. It was an indicator that the significance of CI, CD, and DevOps was not well established in the general public job site. It will take more time for DevOps to be mature and recognized by the broader IT audiences. It will take even longer for DevOps to be established in academia.

4.4 DevOps Professionalism

Having authoritative entities or formally recognized training programs could advance DevOps as a profession. Therefore, we studied whether authorities or training programs exist for DevOps. Many organizations are trying to establish authority, guideline, and standard for DevOps through writing, community building, conferences, and training programs. However, no organization had stands out as the DevOps authority yet.

4.4.1 DevOps Certification To identify DevOps authorities, we searched for DevOps certifications, training programs, communities, events, and authoritative organizations. ICAgile [36] is a front-runner in establishing themselves as the DevOps authority. ICAgile is trying to establish a separated track of DevOps certification, in addition to its Agile tracks. ICAgile built its DevOps learning roadmap through community contribution from pioneers, experts, and trusted advisors. ICAgile depends on experienced DevOps practitioners for guidance. Hence, the primary source of DevOps knowledge is from experience.

The ICAgile's DevOps certification path starts from ICAgile Certified Professional in Foundations of DevOps (ICP-FDO), then ICAgile Certified Professional in Implementing DevOps (ICP-IDO), and finally ICAgile Certified Expert in DevOps (ICE-DO).

When we first looked up the objectives of ICP-FDO in December 2017, there were 22 major topics and 95 sub-topics. The long list of topics and sub-topics shows the complexity of DevOps. No training can cover all 22 major topics and 95 sub-topics thoroughly in the recommended two-day period. Therefore, ICP-FDO provides only an overview of DevOps. Students who received ICP-FDO certificates were not recognized as DevOps practitioners. In April 2018, ICAgile published an official Learning Roadmap DevOps Track [37]. In the new roadmap, ICP-FDO learning objectives have been reduced to five major topics with 14 sub-topics. Hence, even the DevOps experts were in the process of defining and refining DevOps.

ASPE Training [38] was the primary provider for the ICAgile certified ICP-FDO training. The ASPE Training course "DevOps Implementation Boot Camp (ICP-FDO)" [39] actually takes three days, instead of the two days recommended by ICAgile. It takes more time to convey even basic DevOps knowledge than the experts estimate.

In December 2017, objectives for ICP-IDO were not yet defined. Hence, it was hard, even for DevOps experts, to define the learning outcomes needed for DevOps practitioners. In the April 2018 roadmap, there are 4 major topics and 14 sub-topics for ICP-IDO. However, no training organization has figured out how to provide training that satisfies the ICP-IDO learning objectives yet. Therefore, the only way to become a DevOps practitioner is still through hands-on working experience.

4.4.2 DevOps Training Programs For non-technical personnel there are even fewer training programs. In November 2017, we searched for DevOps training in Coursera [40]. When searching Coursera for courses using the keywords "Continuous Integration" and "DevOps", there were 0 results. Hence, DevOps was not even a valid topic in Coursera then.

In July 2018, we searched Coursera using the keywords "Continuous Integration", "Continuous Delivery", and "DevOps" again. The searches returned 22 courses for "Continuous Integration", 16 for "Continuous Delivery", and 13 courses for "DevOps". These searches showed that DevOps gained recognition in eight months between 2017 and 2018. However, none of those courses were actually about CI, CD, or DevOps. Instead, those courses were about Agile, microservices, the cloud, specific tools, and other topics not directly related to DevOps.

In addition, we searched Skillsoft [41] for DevOps courses, since Skillsoft is the global online learning resource for many corporations. Skillsoft returned 32 courses. The search results were similar to Coursera's, with 30 of the 32 courses not directly related to DevOps. Two courses "DevOps Fundamentals: Tools, Technologies, and Infrastructures" and "Integrate Development and Operations (DevOps)" have course hours of one hour and 56 minutes, and one hour and 32 minutes, which provided only a high-level description of DevOps. ICAgile recommends two days for ICP-FDO; Skillsoft's learners could not learn much about DevOps in less than two hours. In conclusion, industry lacks DevOps training for both technical and non-technical personnel.

4.4.3 DevOps Communities, Conferences, and Organizations DevOps is rapidly gaining recognition through the effort of many IT practitioners. DZone Integration Zone [42], DZone DevOps Zone [43], and DevOps.com [44] are significant DevOps communities where people can learn and share information and experiences about DevOps through writing, publications, webinars, and more. DevOps service providers can also market their services in these communities. In addition, there are many personal blog sites about DevOps [45].

Besides the DevOps communities, there are increasing numbers of DevOps conferences and events around the world. All Day DevOps [46] is an annual online conference that runs continuously for 24 hours, with more than one hundred speakers, and audiences from over one hundred countries in 38 time zones. DevOps World - Jenkins World [47] is organized by Jenkins [48], a DevOps tool producer. It provides "opportunities to learn, explore, network, and help shape the future of DevOps and Jenkins" [47]. It has 2500 attendees from all over the globe, and more than one hundred workshops. In addition, DevOps Enterprise Summit [49], started in 2014, has more than 1400 attendees and over one hundred speakers. Finally, there is DevOps Experience Virtual Summit [50], which focuses on experience sharing among DevOps experts. There are many other DevOps events. The many events show that the community is trying to improve DevOps awareness, training, and education by gathering experience from IT practitioners.

In addition to industrial conferences, DevOps is starting to get interest from academic conferences. Centers for Advanced Studies Conference (CASCON) [51], an annual international conference on computer science and software engineering, introduced a CI workshop in 2016 and a DevOps workshop in 2017. International Workshop on Continuous Software Evolution and Delivery (CSED) [52], focuses on CI and CD. It co-located with the 2016 International Conference on Software Engineering (ICSE) [53]. There are also organizations like DevOps Research and Assessment (DORA) [54] that preforms DevOps research, and is trying to institute a "scientific approach to software development, product management, and organizational transformation" [54].

ICAgile defines ICE-DO as those who "gain[ed] exposure in the community by writing great blogs, speaking at conferences, and publishing [DevOps] books" [55]. These communities are important platforms to advance DevOps maturity. In conclusion, DevOps is on its way to become an IT standard. Meanwhile, it heavily relies on IT practitioners' experience, which affirms the potential theory. Nonetheless, the foundation work to formalize DevOps education has started. There are strong communities, conferences, and events where people may learn or publish. Academics also start to show interest in DevOps through scientific research.

4.5 DevOps Practitioners

Since the progress of DevOps heavily relies on the expertise of the DevOps practitioners, we interviewed DevOps experts to learn about their experience.

Table 3 Experts ranked requirements for DevOps

Requirements [0-9]	#1	#2	#3	#4
a. Education?	3	0	5	2
b. Years of hands-on experience?	7	9	8	9
c Diversity of hands-on experience?	9	6	6	9
d. Diversity of skills?	8	5	7	9
e. Knowledge about different tools?	7	3	7	0*

* No ranking provided, but said that tools were irreverent.

4.5.1 Research Process and Result We invited community article writers, conference and webinar speakers, and book authors for interviews. From DZone DevOps Zone [43], we searched for well "liked" DevOps articles and invited their writers for interviews. Eight writers were invited. Then we searched for speakers in All Day DevOps 2017 [46], which had five tracks. We invited keynote speakers and speakers from each track for interviews, 12 in total. DevOps.com [44] offered many DevOps webinars. we invited six webinar speakers for interviews. In addition, we invited two researchers from DORA [54], one instructor from ASPE Training [38], and two DevOps book authors. Last, but not least, we visited the local Python User Group [56] to invite the local IT practitioners to participate.

We presented the proposed grounded theory to the interviewees, then asked the following questions.

1. What do you think about the grounded theory above?
2. How do you become a DevOps expert?
3. What skills should a DevOps expert have?
4. Between 0 and 9, with 0 being least important and 9 being most important, how are these factors contributing to your DevOps expertise?
   a. Education? [0-9]
   b. Years of hands-on experience? [0-9]
   c. Diversity of hands-on experience? [0-9]
   d. Diversity of skills? [0-9]
   e. Knowledge about different tools? [0-9]
5. Others factors important to your DevOps expertise?

In February and March, 2018, we interviewed four volunteer DevOps experts. One interviewee was DevOps book author; one was senior lead instructor of DevOps certification courses; the other two were senior IT practitioners. We transcribed the interviews, and applied selective coding on the transcriptions to verify, reject, or enhance the potential theory. The interviewees provided consistent answers to the questions. Therefore, we determined that there was sufficient data to saturate the theory and seek no addition interviewees.

4.5.2 DevOps Practitioner Requirement Analysis During the interviews, the interviewees ranked the importance of different factors in becoming DevOps experts. Table 3 shows the results. All interviewees consistently ranked education the least important factor of becoming a DevOps practitioner. The interviewees indicated that education could only provide fundamental DevOps knowledge. One of the interviewees said that education was useful "only when youre looking for the first job".

Two interviewees ranked knowledge of different tools low. As one of the interviewees said, "What really matter [was] that [practitioners] have enough experience … understand the logic behind … instead of specific skill." Experienced IT practitioners could learn to use any tools as needed. Another interviewee (#4 in Table 3) did not provide any rank regarding knowledge of different tools. The interviewee considered specific tools were irrelevant to becoming a DevOps practitioner, therefore we assigned zero for the expert. One interviewee ranked knowledge of different tools 7 out of 9, but he remarked that "knowledge in specific tools is not as important as having a broad knowledge of the available tools, [and] how they fit together". As one of the experts said, "Someone having very board experience … [could not] be a huge expert in certain [tool]." Experts claimed that DevOps practitioners should not limit themselves to specific tools, but focus on understanding system level architectural concepts.

All experts ranked either years of hands-on experience, or diversity of hands-on experience the most important factor of becoming DevOps practitioners. One interviewee pondered that DevOps practitioners need more than experience. They need diversified experience, where "diversity needs to represent both breadth of experience and depth of experience". It confirmed the potential theory that the DevOps craftsmanship could only be learned through experience, specifically diversified experience.

4.5.3 DevOps Practitioner Expectation Analysis Responses for the other questions are summarized below.

Question 1: The Interviewees agreed that DevOps was experience based. One of the interviewees quoted from Daman Edwards that "there [was] no one true authority on what DevOps [was] or [wasnt], because DevOps [was] an experience -based movement" [57].

The interviewees also agreed that DevOps was "akin to an art form" for there was no "formula or recipe to accomplish [DevOps] goal". Therefore, DevOps was not craftsmanship that IT practitioners learned, and then applied. Instead, IT practitioners applied their experience to achieve DevOps craftsmanship. One of the interviewees further argued that DevOps was not craftsmanship by a single IT practitioner, but by a team. It was about "creating self-sufficient teams" that were capable of carrying out the craftsmanship. Most IT development involved a team. The forefront of DevOps was to handle the social aspect of the development team.

Question 2: One interviewee concluded that DevOps practitioners should have "the breadth and depth of IT experience that [would] provide context for all of the DevOps practices and tools".

Question 3: None of the interviewees named a specific skill, such as Java programming. One interviewee said, "There [was] not a truly DevOps tool or ... skill." Another interviewee summarized that "DevOps expert must have a wide variety of technical, leadership, management, and customer support skills that [came] from actual hands-on experience in a wide variety of IT roles. But among those skills, none [was] so fundamental that it [was] required". What DevOps practitioners needed were high-level skills, such as problem solving and automation. DevOps did not entail any fixed tool, skill, or solution. Therefore, each DevOps implementation was unique as a form of art, as depicted in the potential theory.

Question 5: One of the interviewees further illustrated that DevOps practitioners need more than skills. DevOps practitioners also needed trust and authority from the higher management. "DevOps [was] about transforming an organization." DevOps implementation often provokes cultural changes and causes fear. IT practitioners need empowerment and trust to overcome the barrier. Therefore, many IT practitioners argue that DevOps is a cultural, procedural, and technological movement [1] [7] [8] [9] [10] [11] [12] [13], with cultural changes overshadowing procedural and technological changes. DevOps experts described DevOps practitioners as individuals with wide and deep hands-on working experience, problem solving skills, and interpersonal skills. In addition to having the right DevOps practitioners, companies wanting to adopt DevOps, should start by embracing the DevOps culture.

4.6 Summary

In conclusion, the GT project studied DevOps education in depth, from academic and industrial perspectives, through questioning, selective coding, surveying, data gathering, and making comparisons. From the study, a GT theory emerged that DevOps education heavily relies on hands-on working experience, while classroom education provides fundamental DevOps knowledge. With a strong DevOps community already existing, DevOps is on its way becoming an IT standard. However, until a DevOps standard is broadly accepted by the IT industry, comprehensive DevOps education is not likely achievable. In the meantime, institutions should include fundamental DevOps culture, procedure, and technology in education, and increase the opportunity for students to have hands-on practice.

5 DevOps Education Hypotheses

The in-depth GT study about DevOps education concludes that academic institutions should provide fundamental DevOps education (in culture, procedure, and technology) to prepare students for their future DevOps advancement. Upon reflection on the outcome, we propose 11 hypotheses in five categories that could ascertain the benefits that academia, students, and industry stand to receive by teaching and adopting DevOps.[1]

5.1 DevOps in Programming Assignments

The first group of hypotheses studies whether embedding different DevOps processes into programming assignments will improve student learning outcomes.

Hypothesis #1: Students who employ the continuous code integration pipeline with auto-build and auto-deploy while doing programming assignments get statistically significant higher grades for their programming assignments.

Hypothesis #2: Students who implement auto-test while doing programming assignments get statistically significant higher grades for their programming assignments.

Hypothesis #3: When doing assignments, students who receive continuous feedback regarding their progress throughout implementing their assignments get statistically significant higher grades than other students who submit their assignments only upon completion.

5.2 DevOps Skills Transferability

The second group of hypothesis studies whether DevOps skills

[1] Because of space limited, inspiration, purpose, and falsification of the hypotheses can be found in the complete report [60].

are transferable.

Hypothesis #4: Students, who practice a combination of continuous code integration, auto-build, auto-deploy, and auto-test in their first two software engineering courses, get statistically significant higher grades in senior-level non-software engineering courses with programming components, including computer networks, database management systems, and computer graphics.

5.3 DevOps Academic Values

The third group of hypotheses studies whether DevOps adds value to academic publications and can initiate joint DevOps research with computer science academics in non-software engineering specializations.

Hypothesis #5: Compared to research papers without artifacts, research papers with "Reproducible" artifacts published in conferences, like Foundations of Software Engineering (FSE) [58], receive statistically significant higher number of citations within the first five years of publication, since the ability to create "Reproducible" artifacts is a major benefit of DevOps.

Hypothesis #6: Students who employ continuous inspection with artificial intelligence while doing programming assignments get statistically significant higher grades for their programming assignments.

5.4 DevOps Employment Opportunities

The fourth group of hypotheses studies whether knowing DevOps improves students' employment opportunities and speeds up job integration.

Hypothesis #7: Students who practice a combination of continuous code integration, auto-build, auto-deploy, and auto-test while doing programming assignments have a higher chance to be employed.

Hypothesis #8: Students who practice a combination of continuous code integration, auto-build, auto-deploy, and auto-test while doing programming assignments value the course more when adapting to industrial work.

5.5 DevOps in Industry

The fifth group of hypotheses is more industry oriented.

Hypothesis #9: IT practitioners with industrial working experience, can learn DevOps skills faster than students without industrial working experience.

Hypothesis #10: Companies adopting DevOps culture (acceptance of development-operations as one team, willingness to fail fast, experiment with new technology, and focus on quality [8]) have a higher employee satisfaction rate then companies only adopting DevOps technology (procedures and tools).

Hypothesis #11: Companies using Agile benefit more from adopting DevOps than companies using non-Agile project management methodologies, in term of key performance factors, such as deployment frequency, deployment lead-time, and mean time to recover.

Studies based on my 11 DevOps education hypotheses could help illuminate the path and direction for DevOps education in academia and industry.

6 Limitations

GT uses open coding to identify key topics of interest from field stakeholder interviews. Some may argue that open coding is a personal interpretation of the interview transcriptions. Therefore, the validity of open coding is questionable. To address this concern, each interview transcription was open coded by two investigators. Through comparison and discussion, we tried our best to eliminate misinterpretation and personal opinion.

If all the interviewees and data are from the same underlying population, a GT project may have external validity problem. To break the localization barrier. we searched the computer science curricula of institutions around the globe; searched for DevOps certification, training, conferences, and communities internationally; invited DevOps experts for interview worldwide.

GT is a qualitative research methodology. Whether a data set is saturated depends on its quality, instead of its quantity. Therefore, the data set will not be statistically significant. The project could have conducted addition interviews to improve its level of confidence, but was restricted by limited resources.

Studying a fast growing, immature topic like DevOps also poses a threat to data reliability. There are new material, concepts, and events added to the pool every day. we had to stay close to all the latest developments and stay up to date.

7 Conclusion

DevOps is a popular movement in the IT industry, which sparks high demand for DevOps practitioners. To prepare computer science students for the demand, we used grounded theory (GT) to study DevOps education from both academic and industrial perspectives.

From the academic perspectives, we found that institutions were not teaching students about DevOps. Academics were not interested in learning or adopting DevOps, therefore it was unlikely for academics to teach DevOps.

From the industrial perspectives, there were high demand for DevOps practitioners. However, industry could not agree on the roles and responsibilities of DevOps practitioners, so they were expected to be Jacks-of-All-Trades. Fortunately, there were strong DevOps communities, with abundant DevOps blogs, webinars, industrial research reports, organizations, conferences, and books to advance DevOps as an IT standard. Industry might agree on a set of DevOps learning objectives in the near future. Meanwhile, DevOps experts agreed that IT practitioners could only learn DevOps through hands-on experience.

The GT project concluded that academic institutions cannot provide comprehensive DevOps training. Instead, academic institutions should provide fundamental DevOps education (in culture, procedure, and technology) to prepare students for their future DevOps advancement, and satisfy the high demand for DevOps practitioners in industry.

In order to encourage DevOps education, we proposed 11 hypotheses, essentially studies anyone could attempt, to establish the benefits academia stands to receive from DevOps education.

REFERENCES

[1] I. Sacolick, "What is devops? Transforming software development," DZone, Inc., 26 02 2018. [Online]. Available: http://www.infoworld.com/article/3215275/devops/what-is-devops-transforming-software-development.html. [Accessed 27 02 2018].

[2] XebiaLabs, Inc., "2017 The Year DevOps Got Real," [Online]. Available: https://xebialabs.com/assets/files/whitepapers/DevOpsGetsReal.pdf. [Accessed 16 03 2018].

[3] D. A. Tamburri and D. Perez-Palacin, "Special Issue: DevOps Quality Engineering," *Journal of Software: Evolution and Process,* pp. 1-3, 2018.

[4] ACM & IEEE, "Computer Science Curricula 2013 - Curriculum Guidelines for Undergraduate Degree Programs in Computer Science," Association for Computing Machinery (ACM) and IEEE Computer Society, 2013.

[5] B. B. N. de Franca, H. J. Jeronimo and G. H. Travassos, "Characterizing DevOps by Hearing Multiple Voices," in *Proceedings of the 30th Brazilian Symposium on Software Engineering (SBES)*, 2016.

[6] F. Erich, C. Amrity and M. Danevaz, "A Qualitative Study of DevOps Usage in Practice," *Journal of Software: Evolution and Process,* 28 06 2017.

[7] W. Jackson, "2017 State of the Software Supply Chain," 2017. [Online]. Available: www.sonatype.com. [Accessed 20 04 2018].

[8] M. Werner, "The 2018 DZone Guide to DevOps Culture and Process Volume V," DZone, Inc., 2018. [Online]. Available: https://dzone.com/storage/assets/7828456-dzone2018-researchguide-devops.pdf. [Accessed 16 01 2018].

[9] M. Schmidt, "DevOps and Continuous Delivery: Not the Same," DevOps.com, MediaOps, LLC., 08 04 2016. [Online]. Available: https://devops.com/devops-and-continuous-delivery-not-same/. [Accessed 12 04 2016].

[10] A. Streets, "Dedicated DevOps? I Smell a Rat…," DZone, Inc., 25 07 2017. [Online]. Available: https://dzone.com/articles/dedicated-devops-i-smell-a-rat. [Accessed 28 07 2017].

[11] J. Lee, "Can DevOps Be a Role?," DZone, Inc., 24 08 2017. [Online]. Available: https://dzone.com/articles/can-devops-be-a-role. [Accessed 30 08 2017].

[12] xMatters & Atlassian, "DevOps Maturity Survey Report 2017," 2017. [Online]. Available: http://info.xmatters.com/rs/178-CPU-592/images/atlassian_devops_survey.pdf. [Accessed 02 09 2017].

[13] A. Wallgre, "You Can't Buy DevOps in a Box," DZone, Inc., 26 08 2017. [Online]. Available: https://dzone.com/articles/you-cant-buy-devops-in-a-box-electric-cloud. [Accessed 26 08 2017].

[14] P. M. Duvall, S. Matas and A. Glover, Continuous Integration: Improving Software Quality and Reducing Risk, Addison-Wesley, 2007.

[15] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, 1999.

[16] J. Humbel and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010.

[17] A. Still and P. Stanhope, DevOps and DNS - Improving Web Application Performance at the DNS Layer, O'Reilly, 2017.

[18] A. Strauss and J. Corbin, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Second ed., Sage Publication, 1998.

[19] B. G. Glaser and A. L. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine Transaction, 1967.

[20] K.-J. Stol, P. Ralph and B. Fitzgerald, "Grounded Theory in Software Engineering Research: A Critical Review and Guidelines," in *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, Austin, Texas, 2016.

[21] S. Adolph, P. Kruchten and W. Hall, "Reconciling perspectives: A grounded theory of how people manage the process of software development," *The Journal of Systems and Software,* vol. 85, pp. 1269-1286, 8 February 2012.

[22] M. Hilton, N. Nelson, T. Tunnell, D. Marinov and D. Dig, "Trade-Offs in CI: Assurance, Security and Flexibility," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, Paderborn, Germany, 2017.

[23] M. Hilton, T. Tunnell, K. Huang, D. Marinov and D. Dig, "Usage, costs, and benefits of continuous integration in open-source projects," in *31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2016.

[24] QS Top Universities, "2017 QS World University Rankings by Subject - Engineering and Technology," [Online]. Available: https://www.topuniversities.com/university-rankings/university-subject-rankings/2017/engineering-technology. [Accessed 03 08 2017].

[25] G. J. Myers, The Art of Software Testing, 1st ed., New York: Wiley, 1979, p. 177.

[26] The Australian National University, "COMP1040 The Craft of Computing," [Online]. Available: https://programsandcourses.anu.edu.au/2017/course/COMP1040. [Accessed 24 08 2017].

[27] University of Texas at Austin, "Curriculum - Department of Computer Science," [Online]. Available: https://www.cs.utexas.edu/undergraduate-program/academics/curriculum. [Accessed 17 08 2017].

[28] M. Werner, "The DZone Guide to DevOps, Continuous Delivery and Automation Volume IV," DZone, Inc., 2017. [Online]. Available: https://dzone.com/storage/assets/4152382-dzone-guidetodevopscontinuousdeliveryautomation.pdf. [Accessed 04 02 2017].

[29] J. Lee, "Are You Qualified to Be a DevOps Engineer?," DZone, Inc., 12 09 2017. [Online]. Available: https://dzone.com/articles/are-you-qualified-to-be-adevops-enginee. [Accessed 15 09 2017].

[30] LinkedIn Corporation, "LinkedIn," [Online]. Available: https://www.linkedin.com/.

[31] C. Delanbanque, "7 Steps to DevOps Hiring Success," DevOps.com, MediaOps, LLC., 21 06 2017. [Online]. Available: https://devops.com/7-steps-devops-hiring-success/. [Accessed 09 01 2018].

[32] Wonster Worldwide, Inc., "Monster," [Online]. Available: https://www.monster.com/jobs.

[33] DZone, Inc., "DZone," [Online]. Available: https://dzone.com/pages/about.

[34] IEEE Computer Society, "IEEE Computer Society Jobs Board," [Online]. Available: https://computer.org/objs.

[35] J. Ambrosio, "DevOps Hiring: Look for the Right Mindset," InformationWeek, 03 10 2018. [Online]. Available: https://www.informationweek.com/devops/devops-hiring-look-for-the-right-mindset/d/d-id/1332949. [Accessed 09 10 2018].

[36] ICAgile, "International Consortium for Agile (ICAgile)," [Online]. Available: https://icagile.com/.

[37] C. Garlick, D. DeGrandis, G. Gotimer and T. Guay, "ICAgile Learning Roadmap DevOps Track," ICAgile, 04 2018. [Online]. Available: https://icagile.com/Portals/0/LO%20PDFs/DevOps%20Learning%20Outcomes.pdf. [Accessed 27 07 2018].

[38] ASPE Training, "ASPE Training," [Online]. Available: http://aspetraining.com/.

[39] ASPE Training, "DevOps Implementation Boot Camp (ICP-FDO)," [Online]. Available: http://aspetraining.com/courses/devops-implementation-boot-camp-icp-fdo. [Accessed 28 07 2018].

[40] Coursera Inc., "Coursera," [Online]. Available: https://www.coursera.org/.

[41] Skillsoft Corporation, "Skillsoft," [Online]. Available: https://www.skillsoft.com/.

[42] DZone, Inc., "DZone Integration Zone - Enterprise Integration news, tutorials, tools," [Online]. Available: https://dzone.com/enterprise-integration-training-tools-news.

[43] DZone, Inc., "DZone DevOps Zone - DevOps news, tutorials, tools & reviews," [Online]. Available: https://dzone.com/devops-tutorials-tools-news.

[44] DevOps.com, MediaOps, LLC., "Where The World Meets DevOps," [Online]. Available: https://devops.com/.

[45] E. Novoseltseva, "Top 16 DevOps Blogs You Should Be Reading," DZone,

Inc., 18 02 2018. [Online]. Available: https://dzone.com/articles/top-16-devops-blogs-you-should-be-reading. [Accessed 21 02 2018].

[46] Sonatype Inc., "All Day DevOps," [Online]. Available: https://www.alldaydevops.com/.

[47] CloudBees Inc., "DevOps World - Jenkins World," [Online]. Available: https://www.cloudbees.com/devops-world.

[48] CloudBees Inc., "Jenkins," [Online]. Available: https://jenkins.io/.

[49] IT Revolution, "DevOps Enterprise Summit (DOES)," [Online]. Available: https://events.itrevolution.com/.

[50] DevOps.com, MediaOps, LLC., "DevOps Experience Virtual Summit," [Online]. Available: https://devopsexperience.io/.

[51] IBM Canada Laboratory, "Centers for Advanced Studies Conference (CASCON)," [Online]. Available: https://www-01.ibm.com/ibm/cas/cascon/index.jsp.

[52] ACM SigSoft, "International Workshop on Continuous Software Evolution and Delivery (CSED)," International Conference on Software Engineering (ICSE), 14 05 2016. [Online]. Available: http://continuous-se.org/CSED2016/.

[53] Association for Computing Machinery (ACM), "International Conference on Software Engineering (ICSE)," [Online]. Available: http://www.icse-conferences.org/.

[54] DevOps Research and Assessment LLC., "DevOps Research and Assessment (DORA)," [Online]. Available: https://devops-research.com/.

[55] ICAgile, "ICAgile Certified Expert In DevOps (ICE-DO)," [Online]. Available: https://icagile.com/Learning-Roadmap/DevOps/Expert-in-DevOps. [Accessed 19 12 2017].

[56] The Edmonton Python User Group, "Edmonton.py," [Online]. Available: http://edmontonpy.com/.

[57] D. Edwards, "The (Short) History of DevOps," 17 09 2012. [Online]. Available: https://www.youtube.com/watch?v=o7-IuYS0iSE. [Accessed 13 04 2019].

[58] ACM, "ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)," [Online]. Available: https://www.esec-fse.org/. [Accessed 02 02 2020].

[59] I. Monster Worldwide, "Monster," [Online]. Available: https://www.monster.com/jobs.

[60] C. Pang, "Grounded Theory for DevOps Education," Ph.D. dissertation, Dept. of Computing Science, University of Alberta, Edmonton, AB, 2019. [Online]. Available: https://doi.org/10.7939/r3-b09b-vd20.