Understanding the Gaps in Software Engineering Education from the Perspective of IT Leaders: A Field Study

Fernando Bona[a], Rafael Chanin[b], Nicolas Nascimento[c] and Afonso Sales[d]
School of Technology, PUCRS, Porto Alegre, RS, Brazil

Keywords: Software Engineering, Software Engineering Education, Skillset in Software Engineering.

Abstract: Teaching software engineering is challenging as it is a field widely desired by the market, and it requires professionals with an increasing set of skills. There are gaps in this learning process perceived both by professionals from the academia and from the industry that need to be further investigated in order to find strategies to reconcile the expectations from both environments. In this sense, this study aims to investigate these gaps from the perspective of Information Technology (IT) team leaders. In order to do so, we chose a qualitative approach to absorb as much knowledge as possible from each interviewed leader by using a field study methodology. Among the results obtained, we identified the appropriate profile to become an IT leader, what IT leaders expect regarding the skills of IT novices, and how the academia and the industry could work together to build an environment that better prepares these professional, according to the view of the interviewed leaders.

1 INTRODUCTION

Software engineering (SE) is the science field which studies the production and management of software. It is split in many subfields, such as software development, requirements engineering and management, software testing and many others. Each of these subfields represents roles performed by industry professionals (Sommerville, 2011). Besides that, the teaching of SE is based on the premisse of an optimal balance between theory and practice, so that students can develop fundamental concepts and knowledge together with skills that aid in the resolution of real world problems that the industry faces (Ouhbi and Pombo, 2020).

There are different techniques employed by teachers to optimize the teaching of IT students, given the constant challenge presented by the industry to the academia that demands learning that is focused on what the industry requires (Bruegge et al., 2015). In this scenario, complementary courses have appeared (Nascimento et al., 2019). These courses do not follow the formal structure of an undergraduate course, and can be taken in parallel with college and focus on specific tech and methodology that is demanded by industry standards.

Some studies have presented indicatives that these complementary courses[1] have a high relevance for students that take them. Steglich *et al.* (Steglich et al., 2020), for example, has identified a 88% course-to-market ratio.

Therefore, the main goal of this research is to *identify the gaps between the formal teaching in the academia and industry demands*. Besides that, we have a secondary goal, which is to *understand how to optimize the formal teaching so it can become closer to the industry*.

Given the aforementioned goals, we have set the following *research questions* (RQ):

- **RQ1:** How is this academia-industry gap in skill development perceived by IT leaders?
- **RQ2:** What actions and mechanisms can be adopted by formal undergraduate programs to get closers to the industry?

Considering the goals and questions presented, we have performed a field study with 25 IT team leaders,

[a] https://orcid.org/0009-0008-7728-3244
[b] https://orcid.org/0000-0002-6293-7419
[c] https://orcid.org/0000-0002-0080-8822
[d] https://orcid.org/0000-0001-6962-3706

[1] A complementary course is usually a course performed either by the industry or industry-academia partnerships covering emerging technologies that are being adopted by the market and allowing students to catch up with the industry.

Bona, F., Chanin, R., Nascimento, N. and Sales, A.
Understanding the Gaps in Software Engineering Education from the Perspective of IT Leaders: A Field Study.
DOI: 10.5220/0011959100003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 2*, pages 511-518
ISBN: 978-989-758-641-5; ISSN: 2184-5026

in which we sought to map the gap between academia and industry from these professional's perspectives. Since they are used to lead novice developers when they reach the industry, we might be able to understand the process of learning that a novice developer requires once inside a company. In this study we will use the terms novice and newcomer interchangeably.

Among the main results obtained by this study, we have found that IT leaders consider that professionals that enter into the market have: i) lack of technical deepness, ii) lack of communication, iii) lack of business knowledge, iv) lack of time management skills, v) lack of patience, vi) lack of resilience, and vii) lack of pro-activity. To deal with this, leaders apply mentorships, feedbacks and collaborative work.

2 BACKGROUND

In this section we depict and explore the concepts that are important to understand this study.

2.1 Software Engineering

Software engineering can be defined in several ways; a standard definition is *a systematic collection of good practices in the context of software development processes* (Mall, 2018). This systematization is based on academic research, as well as in observations from the industry (Mall, 2018).

In this sense, the goal of the field is to discuss the cost-benefit of software development techniques. These techniques support the process since they were conceived by using engineering approaches (Mall, 2018). Beslmeisl *et al.* (Beslmeisl et al., 2016) describe the importance of scientific research in the context of software engineering education, which investigates methods to teach basic as well as advance concepts. The software engineering domain does not focus only on the final product, but in the whole process. Therefore, in order to deliver content that can help professionals in their career, it is important to oversee all aspects (Beslmeisl et al., 2016).

2.2 Software Engineering Education

Many scientific events have gathered researchers to discuss techniques that can better help instructors in the process of teaching software engineering (Cunha et al., 2018). Several challenges have been presented: i) difficulty in engaging students; ii) difficulty in organizing and executing hands on activities; iii) difficulty in finding good books and articles; and iv) difficulty in finding tools to support the learning process (Ouhbi and Pombo, 2020). To solve this, several approaches have been tested by software engineering teachers: Flipped Classroom (Kiat and Kwong, 2014), Problem based learning (Richardson and Delaney, 2009), Challenge based learning (Binder et al., 2017), Project-based Learning (Souza et al., 2019), Role Playing (Zowghi and Paryani, 2003), and Serious Games (Hainey et al., 2011).

2.3 Gaps Between Academia and Industry

The connection between academia and industry is that the first prepares students that will become future professionals in the market. However, it is not uncommon to find gaps in this relationship (Kumar, 2006). This happens for a lot of reasons. One example is that sometimes the academia is not able to keep up with the current market needs, since it changes rapidly.

Therefore, several authors have tried to map out the gaps between academia and industry in order to better understand this phenomena. In this context, Oguz e Oguz (Oguz and Oguz, 2019) have identify the following gaps: i) the software industry is always expanding to new areas; ii) academia does not receive feedback from the industry; iii) software engineering course can not keep up with the new practices in the market; iv) courses are taught individually, as a box, and students sometimes can not see the connection among them; v) instructors are usually not connected with the industry; vi) soft skills are a must in the software engineering context; and vii) undergraduate courses do not follow the current trends of the market.

Moreover, Oguz e Oguz (Oguz and Oguz, 2019) investigated which abilities are most needed in the industry. In regards to hard skills the author mentioned: i) programming; ii) database; iii) understand the software engineering process; and iv) having work experience. When it comes to soft skills: i) being fluent in English; ii) teamwork; iii) problem solving; iv) critical thinking; and v) communication.

3 IT LEADERS FIELD STUDY

3.1 Methodology

Field studies, according to (Singer et al., 2008), seek to investigate participants in some activity and to characterize how these participants deal with them, understanding how they solve problems in a defined context. Data from a field study is usually qualitative. However, data collection can be performed through

observation, through questionnaires or through interviews to obtain data about the participants' activities (Singer et al., 2008).

In this study, we have decided to collect data through interviews which were conducted remotely due to the pandemic (Platto et al., 2020). Thus, we have followed the recommendation proposed by (Seaman, 2008) for semi-structured interviews; even though a list of questions was established, we could be open to other topics expressed by participants.

Initially, two pilot interviews were performed with two professionals from the research field but who had experience leading IT teams in order to improve the quality of the data collection instrument. Data from these interviews were not considered for the study, but it allowed us to adjust the questions so that they could be logically organized. Further, three senior SE researchers have reviewed the protocol, each of which with had 6, 15 and 20 years of experience in the SE field respectively.

In a final version, the interview protocol was composed by 11 questions, with 6 profile questions and 5 research-related questions. The profile (Q1-Q6) and research (Q7-Q11) questions were:

1. What is your company size? How many employees does it have approximately?
2. For how long have you been a part of the company?
3. For how long have you been leading IT teams?
4. What technical characteristics do you consider to be the most important to lead an IT team?
5. What behavioral characteristics do you consider to be the most important to lead an IT team?
6. What is the frequency with which you have contact with newcomers in the tech field?
7. When newcomers arrive, what skills do they usually have and that are extremely important to the working environment?
8. When newcomers arrive, what skills do they usually lack and that are extremely important to the working environment?
9. Regarding these skills that newcomers usually lack, how do they usually deal with this and what are the most assertive strategies?
10. Can you identify mechanics in the academia that aid newcomers to outperform others in the working environment?
11. How do you consider that the formal teaching can aid these professionals to arrive better prepared?

In total, 40 IT leaders were invited, by convenience, to participate in the study, with 25 of these leaders actually participating. To identify the amount of participants in the study, we have used the concept of saturation presented by (Creswell, 1998) which consists in performing a set of interviews and assess them until new data is not obtained anymore. Initially, we have performed 10 interviews and did not reaching saturation. After that, we performed 10 more interviews, still not reaching saturation. Next, we performed 10 more interviews and, once we had analyzed 25 participants data points, we realized that the last five did not add any new evidence (reached saturation) and only deepened already-mapped topics.

These interviews were recorded, always with the authorization from the participants. However, these audio files were only used for the transcription of each interview, preserving the identity and privacy of them. The interviews lasted between 20 and 40 minutes, with an average time of 28 minutes.

Data analysis was performed according to (Bardin, 2004), in which we have analyzed the content of the answers, grouped them by opinion agreement of the participants, taking note of the disagreements when they occurred and extracting as much information as possible from the interviews's transcriptions to answers the selected questions. After that, we created categories for the extracted information, allowing for a categorization of the answers through the participants's point of view.

3.2 Participants Profile Questions

The profile questions were proposed in order to characterize the IT leaders that participated in this study. The first questions (with their respective answers) can be found in Table 1. ID identifies the participants.

To sum up, 13 leaders worked in big size companies, 7 in medium size companies and 5 in small companies. The number of workers varied from 3 (which is a startup) up to multinational companies with over 400,000 employees. Moreover, we interviewed professionals with only one year as a leader as well as others with up to 23 years as IT leaders.

Participants also characterized what they believe to be the most important abilities in a IT leader. This information is presented in Figure 1. As it can be noticed, we group these abilities into for subgroups:
Technical Knowledge of Utilized Technology: IT leaders usually do not act as programmers, but they need to know how the technology works and need to know how to guide the team about these technologies.
Management and Leadership Capacity: It is expected from a leader, beyond technical knowledge, a proper usage of management and leadership skills to keep the commitments that the company demands.

Table 1: IT leaders professional profile.

ID	Company Size	Number of Workers	Years of Service	Year Leading Teams
L1	Big	10.000	9 years	9 years
L2	Big	1.500	1 month	8 years
L3	Small	40	7 years	20 years
L4	Medium	200	1 year	10 years
L5	Big	28.000	3 months	15 years
L6	Big	3.100	3 months	8 years
L7	Big	6.000	2 years	7 years
L8	Medium	300	5 months	4 years
L9	Big	10.000	3 years	6 years
L10	Big	6.000	3 years	1 year
L11	Medium	500	3 months	5 years
L12	Medium	110	7 months	4 years
L13	Big	34.000	1 year	4 years
L14	Medium	120	3 years	20 years
L15	Big	1.500	10 years	6 years
L16	Medium	50.000	2 years	14 years
L17	Small	70	5 months	7 years
L18	Big	400.000	21 years	18 years
L19	Small	3	5 years	5 years
L20	Small	18	5 years	4 years
L21	Medium	300	4 years	9 years
L22	Big	3.000	7 years	6 months
L23	Big	1.500	2 months	2 years
L24	Small	4	1 year	9 years
L25	Big	4.000	31 years	23 years

Team Formation Knowhow: IT leaders need to use the profile of team members, knowing the characteristics and skills that each member possesses in order to design teams where members complement each other.

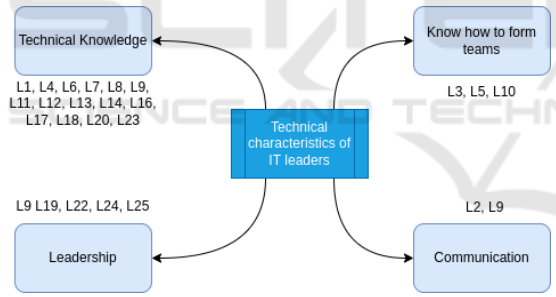

Figure 1: Most important abilities according to the participants.

Communication: Leaders need to be be understood by the team, transmitting information clearly or being transparent whenever possible.

Behavioral skills are expressed in Figure 2, where 12 codes have emerged after summarization:

Active Listening: IT leaders must listen to their teams and gather information even when performing other task within the project.

Communication: Communication between the leader and his/her team needs to be transparent, and everyone must try to understand each other.

Empathy: In some situations it is recommended to exercise empathy within the team, trying to put yourself in someone else's shoes.

Resilience: A leader must persist until it finds a viable alternative. Since the leader is usually the most experience person in the team, he/she needs to set an example.

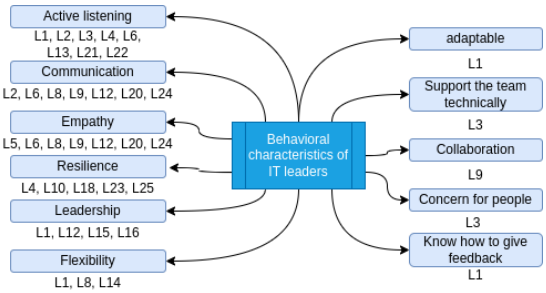

Figure 2: IT leaders most important behavioral characteristics.

Leadership: Not everyone is naturally prepare to become a leader. Most of the times It is necessary to develop this ability throughout the process.

Flexibility: A leader must be flexible, knowing how to talk to the team members and, in some cases, give in to the team members' needs.

Adaptable: Projects constantly change and so must the leader and team be able to cope with these changes.

Technical Support: In many cases, the leader is the most experienced person in the team, even in the technical aspect. So, in difficult situations, the leader needs to guide the learning process to solve these situations.

Collaboration: The leader is part of the team and as such should collaborate and try to ease internal team processes.

Concern for People: The leader must worry with the well-being of his/her team members, ensuring they are comfortable with their roles as this could sometimes depend on the execution of projects.

Knowing How to Provide Feedbacks: The leader, given in many cases possesses the most experience, can and should support the professional development of team members, utilizing feedback to point the good points and the improvement opportunities.

The interviewed leaders were also questioned about when they were used interact with IT newcomers/novices (Figure 3), where 3 codes were identified:

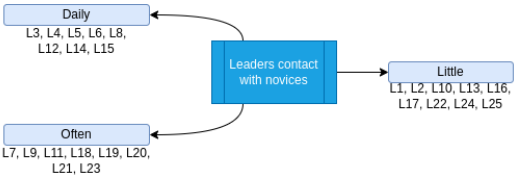

Figure 3: Leaders's interaction with IT newcomers.

Little: Does not possess frequent contact with IT newcomers.

Often: Does not have daily contact, but often interact with IT newcomers.
Daily: Interacts with IT newcomers everyday.

As such, the leader profile tends to be diverse, possessing this characteristics in common, according to the answers to this question, but with some particular ways leader handle their teams.

3.3 Field Study Research Questions

Participants also provided their opinions regarding skills that newcomers have when joining and IT team. The skills that they use to have when they arrive in this team is expressed in Figure 4, where in this question 6 codes emerged after summarization:

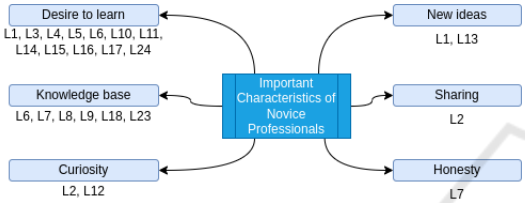

Figure 4: Skills newcomers usually have.

Desire to Learn: Newcomers usually arrive in companies with a desire to learn with industry professionals.
Knowledge Base: The newcomer needs to have a base knowledge, even just programming logic, to be able to trained on the more complex technologies being used by the team.
Curiosity: The newcomer usually questions team members about the usage of certain technologies.
New Ideas: When newcomers arrive from Academia, they usually have seen new technological tendencies and sometimes bring new options for the development team.
Sharing: Newcomers usually have an open mind to knowledge sharing, besides, they benefit widely with knowledge shared by others.
Honesty: A person that is just arriving is usually careful when commenting their limitations or difficulties, generating issues, but that can be avoided with dialogs and honesty with their team colleagues.

The skills that newcomers usually **do not** possess are expressed in Figure 5, where 9 codes have emerged in this question after summarization:
Lack of Technical Depth: Newcomers, as the name suggests, arrive with limited knowledge. This is understandable, and they evolve as they participate more in projects.
Lack of Communication: Some newcomers are shy. However, it is vital to communicate to avoid misunderstandings.

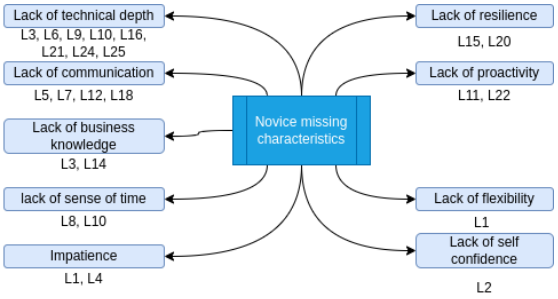

Figure 5: Skills newcomers usually do not possess.

Lack of Business Knowledge: Newcomers tend to think of IT projects just the in term of technical aspects, but not in regards to the business.
Lack of Sense of Time: Another issue with newcomers is related to tasks estimations. Newcomers tend to underestimate the difficulty of the tasks.
Impatience: IT leader have reported that newcomers try to deliver tasks as fast as they can without paying attention to the quality of their work.
Lack of Resilience: IT leaders pointed out that newcomers give up very quickly when they face a harder challenge.
Lack of Proactivity: According to the IT leader, newcomers could be more proactive and take more risks.
Lack of Flexibility: Sometimes newcomers do not understand the company structure and feel frustrated when something can not be done their way.
Lack of Self Confidence: In some situations, newcomers do not fell comfortable to take responsibility since they do not know if they can deliver the given task.

In order to support newcomers, IT leaders came up with a few strategies that are presented in Figure 6.

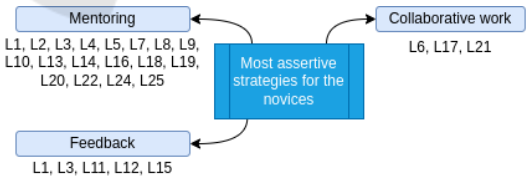

Figure 6: Strategies to help newcomers.

Mentoring: In this process, more experienced team members are selected to partner with newcomers.
***Feedback*:** More experienced team members can provide feedback to newcomers so they can improve their learning process.
Collaborative Work: When working is pairs or in groups, newcomers can develop their abilities faster since they can learn from others.

IT leaders also mentioned what they believe to be effective in the classroom context in order to put stu-

dents more connected to the market (see Figure 7). We identify 10 itens:

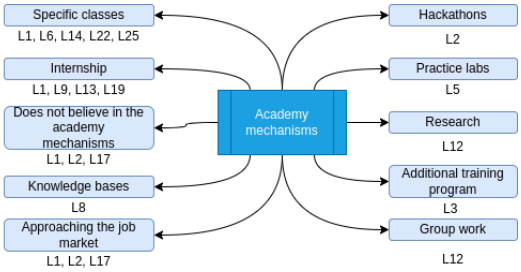

Figure 7: Academia mechanisms to connect students to the market.

Specific Disciplines: Some disciplines are specific to get students close to the industry, making learning closer to the professional reality.
Internship: An effective mechanism is internship, as it inserts the student in the industry. However, the Academia is dependent on the local industry to open this positions.
Labor Market: Some leaders believe that students only learn when they enter the labor market.
Disbelief in the Mechanisms: Some IT leaders do not believe in the Academia mechanisms, considering all of them inefficient to some degree to support the demands.
Knowledge Base: Some leaders point that the ole of the Academia should continue to be just provide basic knowledge and allowing that the industry to perfect the professional.
***Hackathons*:** Events such as *hackathons* have a potential to approximate the student of real scenarios and to create some *networking* with industry professional.
Practical Labs: Some universities offer practical labs in which students take on roles in IT teams, as it is the case of the AGES (Experimental Software Agency) from PUCRS (Yamaguti et al., 2017).
Research: Academia engaging the student to research the industry and its roes to support students' decision-making for what to enter.
Complementary Courses: Some programs offered by the industry in partnership with the Academia in which students are taught members of both institutions (Steglich et al., 2020).
Group Work: In some ways, group work when well-structured help in the development of *soft skills* that are important for the IT labor market.

Leaders also explain the ways in which formal study could support students to reach the market better prepared (Figure 8). Regarding ways in which Academia can support students, 9 codes were identified:

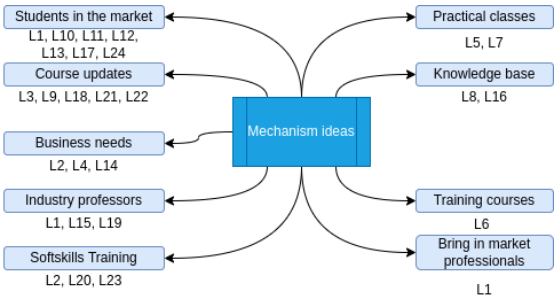

Figure 8: Graduation course strategies to support students.

Students Market Insertion: The Academia needs to seek more partnerships with the industry so that together they can insert students directly in IT industry environments.
Courses Modernization: Some courses are obsolete with the reality being presented by the market, given that the tech fields have a high update frequency of their techniques, methods in which the Academia usually struggles to keep up with.
Business Needs: Students usually possess a decent technical notion in contrast to business notion. These students need to be equally prepared to understand the business ecosystem in which the product he/she builds are inserted.
Market Professionals: The addition of some market professionals as teachers tends to develop some skills before the student joins the job market.
More *Soft skills*: Some courses emphasis too much the technical aspect, with space to add more disciplines that bring personal and professional development to students through *soft skills*.
Practical Classes: Practical classes are usually simulations of what the students may find in the market, if well-implemented.
Knowledge Base: Academia usually prepares students by providing the technical base so that professionals can join the market.
Training Courses: In some specific technologies that are interesting to the student, it can be a good experience to join a training course (Nascimento et al., 2019) about a specific technology.
Bring in Market Professionals: Reducing the distance between Academia and Industry in which members of both parts have an open dialog seeking to reduce these gaps.

As such, through these obtained results, it is possible to understand the position IT leaders have in which they have demands for IT newcomers/novices and try to support them on the other hand, thus, developing the necessary skills to perform in their companies.

4 DISCUSSION

QP1. How Is this Academia-Industry Gap in Skill Development Perceived by IT Leaders? First of all, we have decided to understand what the industry has demanded from these IT novices, in which the IT leaders invited to this study recognize that novice professionals have positive characteristics such as a willingness to learn and a specific knowledge base, but they do not have some skills that would be important to develop for a professional in a way to work in the IT market, such as: *technical depth*, *communication*, *business knowledge*, *sense of time*, *patience*, *resilience*, *pro-activity*, among other skills.

The industry has strategies to overcome these gaps, such as mentorship programs, feedbacks and collaborative work that put pace in the novice professional to understand the logic that the industry demands. However, in order to overcome this gap there are also initiatives recognized by the academy, such as internship programs, specific disciplines/courses and approaches to the job market for students to try to learn their professional assignments in immersive environments of industry, but which still demand considerable efforts from both academia and industry.

IT leaders recommend some strategies such as putting students into the market, modernizing courses, teaching about business needs, classes with market professors, soft skills training, practical classes, focusing on knowledge bases, and training courses and complementary training. These strategies were properly debated by the IT professors, who reported that have their respective interests in bringing students closer to the market, but that in some cases there has been little receptiveness on the part of the industry, especially when looking for projects that have some financial support that motivates their students.

QP2. What Action and Mechanisms Can Be Adopted by Formal 88 Undergraduate Programs to Get Closers to the Industry? Updating courses is a process that takes a consider period of time, since it demands legal procedures with the Ministry of Education, in addition to the fact that technologies are constantly changed in the industry. Teaching about business aspects and soft skills are considered important by teachers, but have seen with some *"strangeness"* by students and some teachers, who think that this knowledge seems to *"blur"* what is really the core of Computing. As a result, teachers tend not to focus on specific technologies but on knowledge bases that reconcile their activities for the development of soft skills and business knowledge.

Finally, the courses offer certain practical subjects, including some optional ones in which students can explore themselves into specific topics, but teachers recognize that formal education cannot be specific to the needs of all companies in the industry.

Therefore, complementary education (*i.e.*, a course performed by partnerships between industry and academia, and sometimes only by the industry) becomes an alternative that can be stimulated by both academia and industry to mitigate this gap between formal education and the demands from the industry, where these courses are offered in different ways and focused on specific technologies that may represent a demand from the industry.

5 THREATS TO VALIDITY

The results found by a field study can sometimes be limited to the interviewees' perspective. Even though we sought to reduce the impacts generated by biases through the opinion of more than one leader, there were somewhat divergent opinions on specific topics in this research. An example of this is how many mechanisms currently exist in the academy for the student formation to the market (where the most part of the leaders believe in these mechanisms), but three of them have do not believe in effectiveness as expressed in Figure 7. Moreover, it was sought through writing to present and elect the results in the most transparent way as possible to protect the understanding of the data as well as the terms or expressions that can be understood as ambiguity by the readers.

6 CONCLUSION

The gaps in the training of IT professionals are challenges for both academia and industry, and common solutions cannot fully resolve this distance that both have between them. In this way, it is essential to provide ways to reduce these gaps, where complementary training courses can be a promising solution: they are relatively faster courses than formal education and focused on a technology or methodology that is highly demanded by the Industry.

Dialogue between academia and industry is fundamental, especially when designing projects such as complementary training courses, which are focused on the needs of the industry in a way that traditional/formal training courses cannot due to several factors, leading in this case to the formal education not being able to handle all technologies and methodologies in the way that the industry wants.

Complementary training courses do not replace the need for formal courses, but tend to explorer deeper into topics that are seen more superficially in the academy, offering more opportunities for practice and increasing students' opportunities to enter the job market. Industry professionals who have taken these courses consider that the impact of additional training courses have been extremely beneficial for their professional careers, and IT leaders recognize that the addition of mechanisms by the academy tends to reduce the gaps in professional training in IT novices.

As future work, we intend to conduct a study with the participation of teachers of formal education (*e.g.*, undergraduate courses), professors of complementary training courses and members of the industry in search of the preparation of a model that can be used by both in the reduction of gaps in professional training between academia and industry.

REFERENCES

Bardin, L. (2004). *Análise de conteúdo. 3ª edição*. Lisboa: Edições.

Beslmeisl, M., Reuter, R., and Mottok, J. (2016). The importance of writing in software engineering education. In *Proc. of the Int. Conf. on Interactive Collaborative Learning*, pages 315–321, Belfast, UK. Springer.

Binder, F. V., Nichols, M., Reinehr, S., and Malucelli, A. (2017). Challenge based learning applied to mobile software development teaching. In *Proc. of the Conf. on Software Engineering Education and Training*, pages 57–64, Savannah, USA. IEEE.

Bruegge, B., Krusche, S., and Alperowitz, L. (2015). Software engineering project courses with industrial clients. *ACM Transactions on Computing Education*, 15(4):1–31.

Creswell, J. W. (1998). *Qualitative Inquiry and Research Design: Choosing Among Five Tradition*. Sage Publications.

Cunha, J. A. O. d., Marques, G. A., Lemos, W. L., Câmara Jr, U. D., and Vasconcellos, F. J. (2018). Software Engineering Education in Brazil: a Mapping Study. In *Proc. of the Brazilian Symposium on Software Engineering*, pages 348–356, São Carlos, Brazil.

Hainey, T., Connolly, T. M., Stansfield, M., and Boyle, E. A. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education*, 56(1):21–35.

Kiat, P. N. and Kwong, Y. T. (2014). The flipped classroom experience. In *Proceedings of the IEEE Conference on Software Engineering Education and Training*, pages 39–43, Klagenfurt, Australia. IEEE.

Kumar, R. (2006). Bridging the gaps–a paradigm shift in software engineering education. Technical report, Indian Institute of Technology Kharagpur.

Mall, R. (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd., Deli, India.

Nascimento, N., Santos, A. R., Sales, A., and Chanin, R. (2019). An investigation of influencing factors when teaching on active learning environments. In *Proc. of the Brazilian Symposium on Software Engineering*, pages 517–522, Salvador, Brazil. ACM.

Oguz, D. and Oguz, K. (2019). Perspectives on the gap between the software industry and the software engineering education. *IEEE Access*, 7(1):117527–117543.

Ouhbi, S. and Pombo, N. (2020). Software engineering education: Challenges and perspectives. In *Proceedings of the IEEE Global Engineering Education Conference*, pages 202–209, Porto, Portugal. IEEE.

Platto, S., Xue, T., and Carafoli, E. (2020). Covid19: an announced pandemic. *Cell Death & Disease*, 11(9):1–13.

Richardson, I. and Delaney, Y. (2009). Problem-based learning in the software engineering classroom. In *Proc. of the Conf. on Software Engineering Education and Training*, pages 174–181, Hyderabad, India. IEEE.

Seaman, C. B. (2008). Qualitative methods. In Shull, F., Singer, J., and Sjøberg, D. I. K., editors, *Guide to Advanced Empirical Software Engineering*, chapter 2, pages 35–62. Springer.

Singer, J., Sim, S. E., and Lethbridge, T. C. (2008). Software engineering data collection for field studies. In Shull, F., Singer, J., and Sjøberg, D. I. K., editors, *Guide to Advanced Empirical Software Engineering*, chapter 1, pages 9–34. Springer.

Sommerville, I. (2011). *Software Engineering*. Addison-Wesley, Boston, USA.

Souza, M., Moreira, R., and Figueiredo, E. (2019). Students perception on the use of project-based learning in software engineering education. In *Proc. of the Brazilian Symposium on Software Engineering*, pages 537–546, Salvador, Brazil.

Steglich, C., Lisboa, A., Prikladnicki, R., Marczak, S., da Costa Móra, M., Olchik, A., Heck, N., Rachid, Y., and Ghidorsi, G. (2020). Agile accelerator program: From industry-academia collaboration to effective agile training. In *Proc. of the Brazilian Symposium on Software Engineering*, pages 21–30, Natal, Brazil. ACM.

Yamaguti, M., de Oliveira, F., Trindade, C., and Dutra, A. (2017). Ages: An interdisciplinary space based on projects for software engineering learning. In *Proc. of the Brazilian Symposium on Software Engineering*, pages 368–373, Fortaleza, Brazil. ACM.

Zowghi, D. and Paryani, S. (2003). Teaching requirements engineering through role playing: Lessons learnt. In *Proc. of the Int. Requirements Engineering Conference*, pages 233–241, Monterey Bay, USA. IEEE.