

Changing the Software Engineers' Self-Efficacy at Industry: A Research Proposal

1st Danilo Monteiro Ribeiro

ZupEDU

ZUP Innovation

Brazil

danilo.ribeiro@zup.com.br

2nd Alberto Souza

ZupEDU

ZUP Innovation

Brazil

alberto.tavares@zup.com.br

3rd Victor Santiago

ZupEDU

ZUP Innovation

Brazil

victor.pinto@zup.com.br

Abstract—Goal: We present a research proposal that aims to understand the relationship between self-efficacy and training in Software Engineering. **Method:** We propose a longitudinal survey with software engineers at Zup Innovation that participating of our bootcamp training. **Expected Results:** We expect to collect data to support our assumptions that self-efficacy can be related to training in Software Engineering. Other assumption is that self-efficacy at the beginning of training is higher than the middle, and that self-efficacy at the end of training is higher than the middle. **Conclusion:** We expect that the study proposed in this article will motivate a discussion about self-efficacy and the importance of training employers in the industry of software development.

Index Terms—Self-efficacy, Training, Research Proposal, Industry Training

I. INTRODUCTION

Bandura defined self-efficacy as a belief in one's capabilities to organize and execute the courses of action required to produce given attainments [?]. In other words, self-efficacy is how much an individual believes that he/she is capable of doing a specific task.

In other areas, self-efficacy is widely studied. For example, Stajkovic and Luthans [?] conducted a meta-analysis to understand the effect of self-efficacy on work-related performance. They found that self-efficacy is related to work performance in sports, education, and health workers.

Self-efficacy determines the amount of effort that the individual expended to cope with a task even if it has impediments and difficulties [?]. An individual who has less self-efficacy tends to avoid difficulties tasks and to put less effort into completing a task [?].

Additionally, if individuals have low self-efficacy and high skill, their performance can be compromised, for example, because they do not believe in their ability to perform the task and can strive less to complete the task. If they have high self-efficacy and low skill, their performance also can be compromised. For example, the task schedule can be mistakenly estimated or the quality of a proposed solution/decision may not be ideal because they are overconfidence about their knowledge [?], [?] [?].

Moore and Chang [?] conducted a study with software engineering students and found that overconfidence self-efficacy is negatively related to performance. Moreover, it was

found that when had an intervention (in this case, performing feedback), self-efficacy was positively related to performing. Therefore, recalibrating self-efficacy becomes important to perform appropriately.

Given that, this paper presents a proposal to evaluate an approach to improve and recalibrate self-efficacy through training (bootcamp) in the software development industry. This approach will be tested at a Brazilian organization called Zup Innovation. Zup Innovation has about 1,500 employers and works only with Information Technology.

In this paper, we briefly introduce the self-efficacy literature, show our research question and hypothesis, our context, how we measure self-efficacy and training, initial results, the expected results and final remarks.

II. BACKGROUND

A. Self-efficacy Theory

As was said previously, self-efficacy can be defined as the belief in one's own ability to accomplish something successfully [?]. Self-efficacy is proposed as a construct of Social Cognitive Theory [?] and also had an own theory purposed by Bandura [?].

Self-efficacy can change behavior and how individuals perceived the reality [?], the central assumption of self-efficacy is that people generally will only attempt things they believe they can accomplish and will not attempt things they believe they will fail [?]. This occurs because individuals generally do not put effort into doing task that they think they cannot. When individuals have a strong sense of efficacy, they believe that they can accomplish even the most difficult tasks. Furthermore, they can see the task as challenges to be mastered, rather than threats to be avoided [?].

Bandura affirms that there are four sources of self-efficacy. The most effective way to impact self-efficacy is mastery experiences [?]. Mastery experiences occur when an individual attempts to do a task, and he/she is successful. This occurs because individuals tend to believe more they are capable to do a task if they did it or did a similar task before, moreover, have done the task well. A famous example of mastery experience is babysitting. Women who have experience taking care of kids before becoming mothers are more confident with their abilities to take care own kids [?].

So, to mastering something, an individual "only" has to practice do the task. However, Bandura [?] presents some important points. For example, if the tasks are easy or very similar to ones already, self-efficacy may be poorly developed. Another point is that the task needs to be difficult but not impossible. The individual needs to attempted difficulty tasks and work through obstacles to improve self-efficacy [?].

Heslin [?] gives another tip to improve mastery experience: breaking down a difficult task into small steps, which are relatively easy, to ensure a high level of initial success, then, given progressively more difficult task in which feedback. Heslin suggests providing opportunities with workshops, training programs, internships, and clinical experiences to improve mastery experiences.

Another factor that influences the perception of self-efficacy is vicarious experience. Vicarious experience can occur when an individual observed others' successes and failures who are similar to him/her. Bandura [?] comments that watching someone like yourself doing and successfully accomplish something you would like to attempt increases self-efficacy. Conversely, when you are observing someone as you fail, your self-efficacy tends to be negatively impacted. However, failure is not always harmful, when as they feel confident, individuals can avoid repeating the errors they observe about others too [?]. Heslin [?] comments that to improve self-efficacy, a manager can do workshops and training sessions, like mastery, because when individuals are watching others in a training session, the manager can provide observational experiences.

The third factor affecting self-efficacy is verbal persuasion. When individuals are persuaded verbally, they are more likely to do the task. Having others verbally support the attainment of a task helps support a person's belief in himself or herself [?]. Verbal persuasion is frequently used by sports coaches to improve self-efficacy [?]. Verbal persuasion builds self-efficacy when managers encourage and praise their competence and ability to improve their performance [?].

The last factor is physical and emotional states that occur when someone contemplates doing something [?]. For example, anxiety, fear, and worry are negatively affecting self-efficacy and can lead to an individual think that cannot perform the tasks [?].

B. Self-efficacy in Software Engineering

Some studies investigated self-efficacy as an antecedent, other as consequent in Software Engineering. We highlight some of them follow. For example, Tsai and Cheng [?] conducted an industrial research that found self-efficacy is related positively to intention to knowledge sharing and knowledge sharing. The authors hypothesized that self-efficacy could improve knowledge sharing at software teams.

Another study investigated that the students' programming self-efficacy beliefs had a strong positive impact on the effort and persistence of Software Engineering students. Also, it was found that self-efficacy is negatively related to seeking help [?]. Arya et al. [?] found that that Self Efficacy is positively but

not very significantly related to organizational commitment. Fu [?] found that professional self-efficacy is positively related to the career commitment of Software Engineers.

Hazzan and Seger [?] identified in their study that high self-efficacy practitioners are:

"more cooperative, have a greater sense of morale working with their team members, feel that their relationships with co-workers are closer, get better managerial support, report higher needs in achievement, dominance, affiliation, and difference and have better attitudes towards change".

Other studies are looking to understand how to improve self-efficacy. For example, Dunlap [?] observed that the use of Problem-Based Learning could improve the self-efficacy of Software Engineering students. Steinmacher et al. [?] found that an online coach called FLOSScoach had a positive influence on open source newcomers' self-efficacy, making newcomers more confident and comfortable during the project contribution process.

Srisupawong et al. [?] revealed that perceptions of autonomy, meaningfulness, and involvement are positively associate with strong self-efficacy. Furthermore, the students' perceptions of vicarious experiences and perceptions of social persuasions demonstrated a positive relationship with self-efficacy. Perceived physiological and affective states demonstrated a negative influence self-efficacy of computer science students.

III. METHOD

A. Methodological Framework

This research proposal comes from a positivist philosophical stances that considers the knowledge must be based on logical inference from a set of observable facts [?]. Positivists are reductionist and studying things by breaking them into simpler components. Positivists prefer methods that start with theories from which verifiable hypotheses can be extracted and tested in isolation [?]. In this research proposal the theory that will guide is Self-efficacy Theory proposed by Bandura [?].

The empirical method selected for this proposal is survey research. Survey research is a research method for collecting information to describe or explain the knowledge, attitudes, and behavior of participants [?].

We intend to use a prospective longitudinal design, that is, studies where the same participants are followed over a period of time. Furthermore, our study is classified as Cohort because all individuals in a defined population are considered over time [?].

TABLE I
RESEARCH CHARACTERISTICS

Philosophical instance	Positivist
Question type	Relational
Research nature	Quantitative
Research method	Survey
Approach	Longitudinal
Panels	Cohort

Table 1 summarizes the characteristics of this research proposal.

B. Research Question and Hypotheses

To conduct this investigation, we have two research questions, first is:

Research Question 1: Is bootcamp training related to perception of self-efficacy in Software Engineering?

As previously exposed in section ??, self-efficacy can be changed in many ways. In this study, we chose to investigate the relationship between self-efficacy and training. Several studies investigate a positive impact of training on self-efficacy in other area like [?], [?] and [?]. Based on this, our first hypothesis is:

H1 There is a relationship between training and self-efficacy in Software Engineering.

The second research question is:

Research Question 2: Are there differences in levels of self-efficacy during training?

Stone [?] says that when individuals believe that they will do similar activities to a previous task that already done, the individuals tend to have high self-efficacy. At our bootcamp, all participants already have experience with software development. So, we believe that at the beginning of training, the self-efficacy's participants is high. However, when individuals have more experience with the activities, they tend to recalibrate their self-efficacy when facing difficulties.

Although they already have experience with software development, the intense training aims to improve their career level. We consider that during the training, participants will face unknown situations and seek new knowledge. Thus, we believe that as new skills are required during training, individuals will tend to have less perception of self-efficacy than at the beginning. However, at the end of the training process, we assume that the individuals' self-efficacy will improve and will be greater than in the middle of the process, as the individuals passed will learn to act as required.

Therefore, we have the following hypotheses:

H2 The level of self-efficacy at the beginning of the training will be higher than in the middle of the training.

H3 The level of self-efficacy at the end of the training will be higher than in the middle of the training.

Another important point is a this investigation is: the beginning of training is the first week, the middle of training is the fourth week and the end of training is eighth week. For each data set, mean scores across all items will be calculated for both beginning, middle, and the end of training. Those mean scores will be compared using a paired samples t-test to examine the significance of the difference between them.

In the next sections, we clarify what is training (bootcamp), what is self-efficacy, and how to evaluate it.

C. How we define training

At this research, training is defined as bootcamp. Bootcamp is an immersive teaching program that focuses on the relevant technical and behavioral skills. Thereby, the programming bootcamp offers hands-on, high-impact teaching to train developers with the requirements desired by the organization.

Individuals who are selected to participate in the bootcamp are hired by Zup innovation to spend three months of intensive training. In the first two months, those selected only train with the content provided by the organization. At third month, those selected are allocated to different development teams on real projects.

At last selection, 933 registrants went through four stages to be selected to our bootcamp. These steps range from code analysis, interviews, and interpersonal assessments. In total, was selected 35 individuals for this bootcamp. Other bootcamps will be created and evaluated.

The main objective of this training is to accelerate learning and increase/recalibrate the self-efficacy of those involved. For this, a team of experts in the technologies used at Zup innovation was recruited to create practical training on it. Zup created on-line training in which individuals can go the whole way on their own. The training difficulty is increasing during the learning.

Experts give one hour of theoretical class per day and two more sessions with two hours a week and are available in a virtual room where all participants must be. They answer questions and about what are the best ways to face certain problems that arise during training. Besides, there are also events such as dojo and pair programming that take place during the training.

At the beginning of each week, a self-efficacy questionnaire is applied so that they can report how much they believe they are capable of in software development activities. More information about the forms will be discussed in next section.

D. How we defined self-efficacy

A self-efficacy scale was developed based on the specifications in the organization's role guide for developers. This role guide was developed by the organization's managers and leaders to assess when a member can update the role level, for example, from junior developer to developer or Senior developer to expert developer. We chose to apply the role developer self-efficacy at bootcamp because the main goal of the training is develop in participants the ability to perform tasks at the developer level.

The role guide has three dimensions: the performance dimension related to how much the member can deliver and supports the team's delivery. The technical domain dimension, which is related to how much the member has ownership over the tools and technology used in the company, and finally, the dimension of supporting culture, which measures how much the members can support the organization's culture. The scale

created from the organization's role guide is called Role Self-efficacy.

The scale was developed on Bandura's proposal to create questionnaires on self-efficacy [?]. All items use a Likert scale 1 to 10. Examples of items are as follow:

- I am able to positively impact the quality of the software.
- I am able to produce code that is easy to maintain and read for other zuppers.
- I am able to produce efficient and scalable solutions.

IV. INITIAL RESULTS

The main objective of this paper is get feedback about our idea and process. However, at this section, we present some initial results.

The self-efficacy scale used in the bootcamp was applied to 108 Zup innovation employers, and an exploratory factor analysis was carried out. Kaisor-Meyer-Olkin (KMO) and Bartlett's sphericity test were used to measure the sampling adequacy. The results showed that the KMO value was 0.947, and the significance of Bartlett's sphericity was 0.000. ($\chi^2 = 2915,422$, $df = 1,128$, $p = 0.000$), indicating that the samples met the criteria for factor analysis purpose by Hair et al [?].

Principal component factor analysis was performed using Oblimax rotation with Kaiser Normalization. Factor analysis yielded a 3-factor solution with an explained variance of 73.81

Furthermore, We also used Cronbach's alpha to measure our scale reliability. We found $\beta = 0,953$ at performance dimension, we found $\beta = 0.957$ at domain dimension and for culture dimension, we found $\beta = 0.858$. According to Streiner [?], values above 0.7 are considered with good reliability. In future works, we intend to present the whole process of scale's validation with more detail. Moreover we pretend to apply Confirmatory factor analysis to our scale too.

Another initial result is that during one of the training sessions, the participants spontaneously commented to the experts that after participating in two weeks of training, that in the beginning, they gave higher values because they believed that were better able to perform the tasks described in the questionnaire, however, after training, they believe that need to improve to reach the level that they believed to have. It is worth mentioning that individuals know that they are participating in research about self-efficacy, but they do not know the main objective.

V. EXPECT RESULTS AND FINAL REMARKS

As mentioned above, self-efficacy can be related to performance and can be impacted by training. Therefore, this work aims to clarify the importance of developing self-efficacy in Software Engineering. The main contribution of the study proposed in this paper is the identification of the relationship between training (bootcamp) and self-efficacy. Moreover, we pretend to understand how self-efficacy change during the weeks of training. We believe that organizations should train and develop individuals' skills and their self-efficacy to perform better.

Other relationships can be investigated in future works. For example, the relationship between self-efficacy and performance in an industrial context. This relationship already was investigate in academic setting [?] in Software Engineering and is very important.

Finally, this research should be a important step towards the consolidation of self-efficacy and training in industrial context in Software Engineering research.