



Investigating the Newly Graduated Students  
Experience after University

Fredrik Olsson — Max Karlson

21th of May, 2019

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the bachelor degree in Software Engineering. The thesis is equivalent to 10 weeks of full-time studies.

**Contact Information:**

Author:

Fredrik Olsson

E-mail: [fredde665@gmail.com](mailto:fredde665@gmail.com)

Max Karlson

E-mail: [maaxkarlson@live.se](mailto:maaxkarlson@live.se)

University Advisor:

Deepika Badampudi

E-mail: [deepika.badampudi@bth.se](mailto:deepika.badampudi@bth.se)

Department of Software Engineering  
Blekinge Institute of Technology  
SE - 371 79 Karlskrona, Sweden

Internet: [www.bth.se](http://www.bth.se)  
Phone: +46 455 38 50 00  
Fax: +46 455 38 50 57

## Abstract

Today's labor market is teeming with software development jobs, and employees are needed more than ever. With this statement, one would believe it is easy for a newly graduated student to start their career. However, according to several studies, there are specific areas where newly graduated Software Engineering students struggle when beginning their first job. Currently, there is a displacement about what the school should focus on when teaching their students. This causes various challenges to arise for newly graduated students when they are initially starting their career. To address this issue, this study aims to identify whether or not there exists a gap between the education provided by the universities, and what is expected from the industry. In accordance with this, the purpose is also the point out which areas might be challenging for newly graduated students, and highlight how the school and industry can benefit from the results of this study.

By conducting interviews with both newly graduated student with one to three years working experience or personnel responsible for hiring new employees at companies, this study will give an insight on which common areas newly graduates may struggle with. Although the result specifies several areas which are challenging to newly graduated students. The greatest challenges which the newly graduated graduated students faced were areas revolving around soft skills. This was in accordance with the opinions of the recruiters. Insinuating that these areas are what the school should focus more on. Other differences between the newly graduated interviewee's opinions and the recruiters are also highlighted in the report Several subjects in school could improve its way of teaching. Furthermore, there are possibilities for companies to better adjust their on-boarding of newly graduated. By addressing the challenges which newly graduated face they can provide their new employees with a better understanding of how to properly work and function in the industry today.

**Keywords:** Software Engineering Education; Software Engineering Challenges; Software Engineer Skills; Software Engineering Curriculum

# Abbreviations

SLR	Systematic Literature Review
SE	Software Engineering
SWEBOK	Software Engineering body of knowledge
KA	Knowledge Area
BTH	Blekinge Institute of Technology
RQ	Research Question
ISO	International Organization for Standardization

# TABLE OF CONTENTS

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	Research Questions . . . . .	4
<b>2</b>	<b>Background</b>	<b>5</b>
<b>3</b>	<b>Method</b>	<b>6</b>
3.1	Selection Groups . . . . .	6
3.2	Interview Structure . . . . .	7
3.2.1	Interviewing a recruiter . . . . .	8
3.2.2	Interviewing an employee . . . . .	9
3.2.3	How will the interviews help this study? . . . . .	9
<b>4</b>	<b>Literature Review</b>	<b>10</b>
4.1	Other studies results . . . . .	10
4.2	SWEBOK . . . . .	11
4.2.1	Software Construction . . . . .	12
4.2.2	Computing Foundations . . . . .	12
4.2.3	Software Engineering Economics . . . . .	12
<b>5</b>	<b>Analysis</b>	<b>14</b>
5.1	Employees . . . . .	14
5.1.1	Challenges . . . . .	14
5.1.2	Strengths . . . . .	17
5.1.3	Improvements . . . . .	18
5.2	Recruiter . . . . .	18
5.2.1	What the recruiters wanted . . . . .	18
5.2.2	What the students were struggling with . . . . .	19
5.3	Their concatenated opinions . . . . .	19
5.3.1	What about code? . . . . .	19
5.3.2	Regarding end-to-end perception . . . . .	20
5.3.3	When it comes to SE Economics . . . . .	20
5.3.4	In the Professional Practice . . . . .	20
<b>6</b>	<b>Results</b>	<b>22</b>
6.1	Research Question 1 . . . . .	22
6.2	Research Question 2 . . . . .	22
6.3	Research Question 3 . . . . .	23
6.3.1	How can the school use this study? . . . . .	23
6.3.2	What about the companies? . . . . .	24

<b>7</b>	<b>Discussion</b>	<b>25</b>
7.1	Comparison to other studies . . . . .	25
7.1.1	Most important skills . . . . .	25
7.1.2	Least important skills . . . . .	25
7.1.3	Largest gap, and most important skill . . . . .	26
7.1.4	Largest gap, but least important . . . . .	26
7.1.5	Smallest gap and most important . . . . .	26
7.1.6	Smallest gap and least important . . . . .	27
7.1.7	Soft Skills . . . . .	27
7.2	How can the results be used . . . . .	27
7.2.1	What did the students like about their education? . . . . .	27
7.2.2	Improvements desired by the students . . . . .	28
7.3	Suggestions to school . . . . .	29
7.3.1	Version control . . . . .	29
7.3.2	Testing . . . . .	29
7.3.3	Modern technologies and Frameworks . . . . .	30
7.3.4	Real life experience . . . . .	30
7.3.5	Quality . . . . .	30
7.4	The limitations of the study . . . . .	30
7.5	Validity Threats . . . . .	31
<b>8</b>	<b>Conclusion</b>	<b>32</b>

## CHAPTER 1

---

### Introduction

*”Is the school providing me with adequate education for this line of work...”*

The sentence above was one of the questions frequently brought to mind by both the authors of this paper. In today’s world, technology and the tools which are used in the industry are constantly changing, a lot of things have to go together in order for what the school is teaching to be of actual value. Begel and Simon mentions in their study that Students who start their professional career have to learn new skills, techniques and procedures, making them novices all over again. The academia and universities has for a long period of time kept up with the changes made in the software development industry.[1, p. 4].

Programming languages and frameworks are just two things a Software Engineer comes in contact with which belongs to this vast concept which is called technology. Blekinge Institute of Technology (BTH) is a school that states that *”The purpose of your education at BTH is so that you can create a career in relation to the subject you have chosen”*[2]. This is one of the reasons why it is important to highlight and verify the quality of the education which is provided by the school. In addition to this, it is highly important to identify the areas where most newly graduated students struggle in order to allow the school to accommodate their teaching accordingly. Radermacher mentions in a study conducted in 2012 that the five most common areas where deficiencies exists are Software Testing, Programming Ability, Teamwork, Oral communication and written communication.[3, p. 22].

### 1.1 Purpose

By conducting this study and comparing the results the information gathered may help shorten the bridge between the universities and the industry, by highlighting important areas which newly graduated Software Engineering students either struggle with or are comfortable in. With the results of this study, universities will be able to educate their students to be prepared for the eventual challenges that newly graduated students encounter today. This study will also aid the companies with their on-boarding of new employees who are directly coming out of a Software Engineering educational program.

### 1.2 Scope

Due to limited time and resources, all of the interviews which were conducted were done in Karlskrona, Sweden. However, the results will still be relevant for all Swedish universities. The questions is not bound to a specific school, but

focuses more on what skills are important. The companies and employees who were contacted mostly consisted of people who had a background in one of the following programs at Blekinge Institute of Technology

- Software Engineering
- Web Programming
- Civil Engineering in Computer Science

### 1.3 Research Questions

Following are the three questions that this paper will answer to:

1. Are there any software engineering areas where the recently graduated software engineering students frequently struggle when beginning their careers in the industry today?
2. Do new employees experience other challenges than recruiters believe new graduates have? If yes, which areas differ?
3. Based on the result from the two other research questions from this study, does a knowledge gap exist between the university and the industry? If yes, what actions can universities and companies take in order to address the gap?

## CHAPTER 2

---

### Background

According to several reports referenced in a systematic literature review (SLR) Garousi et al. suggest that many recently graduated Software Engineering student's struggles during the beginning of their careers [4]. The content of the SLR consisted of the findings from 33 different studies from 12 different countries and highlighted a knowledge gap which existed between the university and the newly graduated students first initial workplace. Due to the diversity in the list of Software Engineering (SE) topics used in the studies, the SLR used the current latest version of SWEBOK at that time, version 3.0. and mapped the SE topics discussed in the papers to the 15 knowledge areas of SWEBOK [5]. SWEBOK stands for *Software Engineering Body of Knowledge* specifies and categorizes different SE knowledge's into different knowledge areas (KA). None of the 33 studies referenced in the paper were conducted in Sweden.

In order to address this, a study was conducted emphasized on highlighting and possibly identifying the existence of a knowledge gap in Sweden. A total of 13 interviews were conducted with the participants either being newly graduated employees or personnel responsible for hiring. The results of the interviews were then summarized and the opinions of the interviewees were then compared to the findings of the SLR. The topics which existed in the result which did not possess any risk of clouding the opinions provided by the interviewees were mapped to the knowledge areas of SWEBOK similarly to how the results were gathered for the SLR.

## CHAPTER 3

---

### Method

This paper will use qualitative data from interviews conducted in Karlskrona. Both of the authors of this study have conducted all the interviews together, one author acted as the interviewer while the other author recorded notes during the entire interview. Due to the interview following a semi-structured concept and having prepared questions, it is easier to advance an interview. Along with the prepared question, the interview structure allowed to ask supplementary questions. The reason why this study used interviews instead of a survey is that a semi-structured interview enables the ability to ask supplementary questions to the interviewees' answers [6, p. 121]. When gathering quantitative data, there is a risk of only scratching the surface of what the respondent can give regarding the area [7, p. 34]. When a respondent sits down for an interview, they usually have an easier time expressing themselves. This gives rise to more information than what you were looking for, both in a good and a bad way. The respondent sometimes drifts away to other topics, so it is necessary as an interviewer to keep the interviewee on track. Most of the time, the respondent share more data about the subject during an interview, and this is usually the information which gets lost if a survey is conducted.

For the sake of this research, in order to reach deeper and get a better understanding, conducting interviews was the better choice. The qualitative data is more beneficial to this study. When coming up with questions to ask respondents that were relevant in the area of research, the decision that a structured interview would not suffice was decided. It is too easy to miss out on information if follow-up questions would not be allowed, which is why all of the interviews follow a semi-structured concept. The downside of using this method is that there might already be biased thoughts founded by the researchers. Their understanding of theories to the design of the interview questions can, therefore, become a disadvantage for getting a reliable answer. This will lead to limited opportunities for the respondents to give their own perspective [8, p. 143].

### 3.1 Selection Groups

When selecting participants for this paper, the authors got in contact with companies and asked if they were willing to participate in this study and whether they had some personnel that fitted our selection groups. All the participants were chosen by a representative of the company who judged which ones were appropriate for this study, which the authors later took contact with to ask if they were interested or not. All participants were selected through a stratified sampling with a snowball sampling instead. Stratified sampling is a way of categorizing your target population into categories, in this case, it is newly graduated employees and recruiters at Software Engineering companies [7, p.

77–78]. The respondents were chosen by a contact person on the respective companies, which then could be contacted regarding an interview. Prior to the recommendation of interviewees, there was no contact with the candidates.

To collect relevant data, there are two selection groups. One of the groups being newly graduated students with a work experience of one to three years in a relevant area of work. This group have had time to start their career and has the highest probability to remember any challenges that might have been present, or is still active, when first starting their career. They have also had time to gather knowledge regarding the different tools that are being used in the industry and are able to draw parallels to their lives as students and what they were being taught. The second selection group is recruiters in this line of business. They were not required to have a strong Software Engineering background but needed some experience dealing with related tasks. The recruiter is responsible for taking in good employees and hence is looking for some specific skills. The recruiter is responsible to take in good employees and is, therefore, looking for some specific skills. Even though the recruiter might not be the one who makes the decision regarding the technical skills, the recruiter is sure to have a piece of knowledge in what to look for. By interviewing the people that are responsible for taking in new personnel, this paper will gain an insight into what the industry looks after in a student, and what is expected of the student. These two groups will show the side of a newly employed student which knows the eventual differences between the school and the industry, and the recruiters will present their requirements for how they want a newly graduated student to be.

Within the time frame of this study, interviews with employees were conducted at three different companies, with a total of 13 participators. 4 of the participants were of the recruiter selection group, and 9 from the newly graduated selection group. It was easier to find newly graduated than recruiters, which explains why there are more of that specific group.

## 3.2 Interview Structure

All interviews were conducted in Swedish. Along with computer written notes for each question done by the researcher not asking the questions, the entire interviews were recorded upon getting acceptance from the respondent. The introduction brought up the four ethical requirements which the interview followed, meaning that the introduction explained the purpose of the study, while also stating the rights of the respondent [9, p. 40]. All the data collected during research is saved anonymously and is inaccessible to the public. The data will only be used by the researchers for the purpose of this study. Together with an introduction and a contract, the interview was divided into two parts, the first part is questions regarding the interviewees' background that is relevant to this study, while the second part covers comparisons regarding results of similar studies. After reading the introduction, the recording began while stating the terms of the contract. The respondent gets asked to accept the terms verbally and if the interviewee accepts the terms, the interview begins. In the first section, the interviewee is asked to share their experience with work from

their previous school-life. By asking the respondent about their experiences from school to working life, it is possible to find out if there were any common patterns, both when it comes to challenges and strengths that may have come from school. Leading to the second part of the interview, the respondent was asked to give their opinion regarding the results of similar studies. This part was based on the SLR [4] which is closely related to our work. Based on the respondents' answer, the study will be compared to the SLR, which would clarify if the results from the previous studies are still relevant today in Sweden. Both these parts will help give an insight into how school life translates into work life. Questions in the first part of the questionnaire bring up struggles during the respondents first job and also asks what the respondent felt the school did right when paralleling to work life. Two terms that were frequently brought up during the research was "Soft Skills" and "Hard Skills". Soft skills refer to more social skills such as:

- Communication
- Working in teams
- Understanding customer/user needs
- Being humble and sympathetic
- SWEBOK definition of Professional Practice

Whilst hard skills were more about knowledge-based skills such as:

- Specific programming language
- Problem solving with programming
- Development related tools such as Git

Because there was a lot of talk about these two areas, this study will also be covering these subjects. Begel and Simon bring up that several newly graduated students have good capabilities when it comes to hard skills. But when it comes to soft skills, a greater deal of students had some problems [10]. Based on whether the person of the interview was a recruiter or an employee, the interview had two different initial parts containing distinct questions. This is to align the interview to the competence of the respondent.

### 3.2.1 Interviewing a recruiter

A recruiter is one of the key-individuals when it comes to the process of choosing new recruits. Therefore, they are obliged to know what they are looking for in an employee. Recruiters additionally possess experience regarding the areas recently graduated Software Engineering students regularly struggles with. This gives direct insight into what the companies want from a newly graduated student. With this said, the interview with a recruiter had questions more aimed towards what to look for in an employee. For example, one question was formulated as follows:

*What do you look for in a potential employee?*

This question is open-ended which gives light to a broad spectrum of answers. Because none of the authors has worked with recruiting before, an open-ended question such as this is necessary. Along with the goal to get a view of what the companies want from their employees, a question was asked to the recruiters to understand what they sought in a newly graduated student.

By getting the idea of how a company thinks and reacts when handling new recruits directly from school, this study will align the word of the recruiters with data from the actual employees and their experiences. This questionnaire also brought up the soft skills and the hard skills to see how a recruiter prioritized the different skills.

### **3.2.2 Interviewing an employee**

The interview for employees had more focus on their step to work from school. The reason for asking these questions to employees was to be able to identify if the respondents have similar issues when leaving school. If for example there is a common trait amongst all of the respondents, there is a reason for us to analyze that subject. Being that these individuals have gone to a working environment after they have completed school, these are the perfect respondents for the research. It is not always what the companies want that the school should teach, but instead a subset of the most commonly used tools. By speaking directly to these respondents, they can give their version of school and what parts they felt missing in their education.

### **3.2.3 How will the interviews help this study?**

The entire interview structure was aimed to gather relevant information from the respondent which could be used to answer the research questions. When asking the newly graduated about their strengths, struggles and what they want to improve on, there would hopefully be a common trait amongst the answer, and that commonality may be something that the school might require to address, which in turn answers RQ1. To follow up this, there is also a question about their school time, and whether they feel there is something that the school did well or did not do so well. By asking both the newly graduated and the recruiters what the the newly graduated may struggle with, it is possible to get a picture if there is one or more common areas of struggle. This is directly connected with RQ2, which tries to identify if there are any apparent areas that requires attention from school. When RQ1 and RQ2 has been answered, it is possible to answer RQ3. The combined results will be analysed to properly answer the last research question

## CHAPTER 4

---

### Literature Review

One of the references used for this study is a compiled study of 33 different papers. These papers reach from 1998 to 2018 and so catches a big time span. All these different studies have been made in different countries and areas, but none in Sweden. The study mentioned contains a summary of all the papers together and a compiled excel sheet which contains notable information from each paper. All these sources are also available in the systematic literature review. Garousi et al.'s study claim that recent Software Engineering graduates often face difficulties when beginning their professional careers [4]. The largest reasons stated is misalignment from university education compared to what the industry uses and needs. One of the areas where the students struggled with most was testing, along with the recommendations that more training and education is needed in this area, both from the university and in the industry. When examining BTH's curriculum for the Software Engineering program, there is one course for testing [11]. Testing is not mentioned in any of the other courses and is also placed quite late in education. But hard skills such as programming, testing, etc. is not enough to tackle the real life of the industry. According to 24 of the 33 studies, the importance to possess soft skills is necessary. Other studies also talk about the importance of soft skills, which is brought up under the 'Other Studies Results' section.

### 4.1 Other studies results

The most commonly noted problem students suffered from was the fact that they lacked soft skills. As previously mentioned, soft skills refer to skills such as communication, collaboration, etc. Begel and Simon show in their study that there were no problems with the students programming skills [10]. Instead,

Some educational recommendations compiled from these different studies mentioned that school required to lessen their emphasis on some mathematical subjects. One study brings up the needs to change the mathematical focus to be more Software Engineering relevant [12]. For example to shift the emphasis from continuous mathematics towards discrete mathematics, probability, and statistics. Another study made by Aasheim et al. claims there is a need to add more soft skills subjects with an emphasis on communication, professionalism and working in teams [13]. Liebenberg et al., speaks of the same improvements to the university [14]. They recommend an emphasis on the following subjects to improve today's curriculum:

- Real-life projects where practical experience is included.
- Soft skills and business skills where the university is not only focusing on technical aspects but also includes these soft skills.
- Be up to date with today's moving technology and still be relevant.

One of the sources to Garousi et al.'s study mentioned one interesting method on how the school would be able to apply the previously mentioned suggestions, which Garousi et al. brought up [4].

*“Instead of a greenfield project, a more valuable experience would provide students a large pre-existing code-base to which they must fix bugs (injected or real) and write additional features. Also valuable would be a management component, where students must interact with more experienced colleagues (students who have taken the class previously, who can act as mentors) or project managers (teaching assistants) who teach them about the code-base, challenge them to solve bugs several times until the “right” fix is found, or who give them sometimes capricious and cryptic weekly commandments on requirements or testing that they must puzzle out and solve together as a team”.*

This method was one of the comparisons which were proposed to our respondents to see if they found the idea suitable for a school environment. As an accompaniment to that question, a follow-up question asked if they would appreciate such a course when they went to school themselves, based on their current experience with working life.

## 4.2 SWEBOK

This study will use the SWEBOK definition of different knowledge areas. SWEBOK stands for *Software Engineering Body of Knowledge* and is an international standard for specifying the different areas of Software Engineering. There are 15 knowledge areas (KAs) in total for the version of SWEBOK this study uses. These KAs are:

- Software Requirements
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process
- Software Engineering Models and Methods
- Software Quality
- Software Engineering Professional Practice
- Software Engineering Economics
- Computing Foundations

- Mathematical Foundations
- Engineering Foundations

This study will focus on the following areas because they suited our interviews the best. A description of the KA's will follow:

#### **4.2.1 Software Construction**

The Software Construction area includes topics related to the development of software programs that will satisfy their requirements and design constraints. This knowledge area covers:

- Software construction fundamentals
- Managing software construction
- Construction technologies
- Practical considerations
- Software construction tools

#### **4.2.2 Computing Foundations**

The Computing Foundations knowledge area covers fundamental topics that provide the computing background necessary for the practice of Software Engineering. Topics covered include:

- Problem-solving techniques
- Abstraction
- Algorithms and complexity
- Programming fundamentals
- The basics of parallel and distributed computing
- Computer organization
- Operating systems
- Network communication

#### **4.2.3 Software Engineering Economics**

The Software Engineering Economics knowledge area is concerned with making decisions within the business context to align technical decisions with the business goals of an organization. Topics covered includes:

- Fundamentals of Software Engineering economics
  - Proposals
  - Cash flow
  - The time-value of money

- Planning horizons
- Inflation
- Depreciation
- Replacement and retirement decisions
- Not for-profit decision-making
  - Cost-benefit analysis
  - Optimization analysis
- Estimation, economic risk and uncertainty
  - Estimation techniques
  - Decisions under risk and uncertainty
- Multiple attribute decision making
  - Value and measurement scales
  - Compensatory techniques
  - Non-compensatory techniques

During the analysis phase, the data has been mapped to one of these areas to maintain a certain scope of a statement. By using this international standard to categorize the answers, this study can be used for comparison even in the future. Another advantage is that it will not be as ambiguous for the reader if the study had defined its own categories. It is also beneficial for this paper, as it makes it easy to compare the study with other similar studies, that also uses this standard. Because the largest source to this study also uses SWEBOK to categorize its results, it was inevitable not to use the ISO. This means that this study can more easily be compared and measured against other studies who also uses the SWEBOK knowledge areas.

## CHAPTER 5

---

### Analysis

When all the interviews were conducted, the data was compiled using a thematic analysis model [15]. The concept of thematic analysis is to find a pattern or a theme. In this case, a theme extracted from our interviews. This theme will then later be examined to see if it is relevant for the study. The reason why this study uses a thematic analysis is that it is an easy way to conclude a result. It is also a flexible method, meaning that it does not have strict rules to it which suits our semi-structured interviews[15]. In an interview, two respondents can answer the same question in two different ways but mean the same thing, which is one variable to take in consideration when choosing an analysis method.

The initial step which was taken when analyzing our results was identifying specific codes from our interviews. The codes were then iterated through and each code was assigned to a theme. All themes were then refined and summarized in order to minimize the total number of themes which existed between all interviews. The last step was to map the different themes which were applicable to the knowledge areas of SWEBOK [5]. By reading the different KA chapters, the results were mapped accordingly to each KAs description. The frequency of each area was determined by the number of times unique respondents mentioned the topic.

It is important to note that all the interviews have been translated from Swedish to English for the sake of this paper.

### 5.1 Employees

Almost all employees had a finished degree in an area related to Software Engineering. After going through all the data, the results were divided into different sections. These sections are:

- Challenges
- Strengths
- Improvements

#### 5.1.1 Challenges

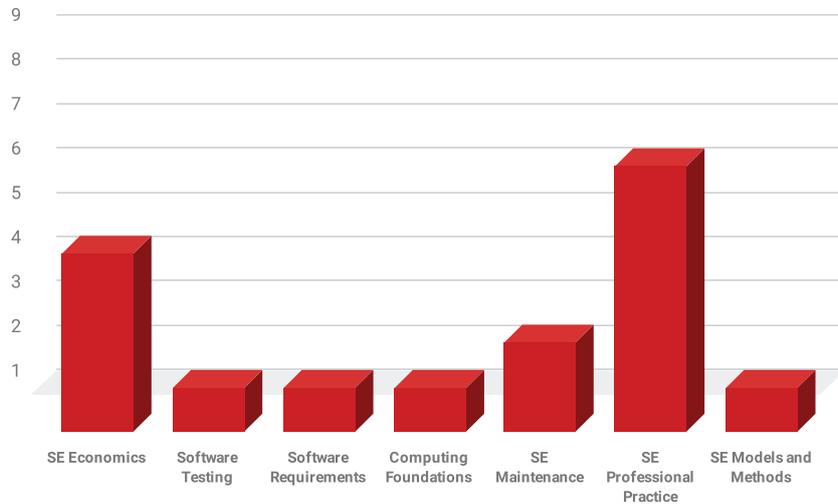
In the questionnaire towards the newly graduated, there was a section which aimed at identifying eventual challenges that respondents may have experienced when they started working. During the interviews, several themes were mapped into matching the SWEBOK knowledge areas [5]. Some themes which were brought up during the interview could not be mapped to a SWEBOK area, because it would lose importance and meaning when mapped together with an area. Therefore, there are some themes not mentioned as a knowledge area.

Six out of the nine respondents named some specific soft areas as being challenging during the initial start of their career. One area was related to being able to take constructive criticism. This occurred mainly during the respondents daily work during phases of code reviews which was done in conjunction to pull requests. Another soft skill which was mentioned was cooperation. Some of the interviewees said that there should be an emphasis put on understanding the concept of money in Software Engineering. One interviewee, in particular, brought up that in some situations you're actually working with someone's life savings and it's important to be cautious with the time you take from them. This is a concept which gets more important the more experience the employee gains, as with more experience generally there will be more responsibilities.

When the question regarding challenges was asked directly, few of the individuals pointed out that maintenance was something they found challenging. However, when maintenance was brought up during other parts of the interview, it was talked as a significant area and something they lacked experience in. This was especially true during the start of their career. Some of the respondents also mentioned that maintenance is something which was done on a daily basis in some way. This also led to the interviewee highlighting the importance of writing clear code. According to the respondents, a lot of their daily work goes into maintaining, reading and incorporating already existing code into what they are doing themselves.

A challenge which was commented from one individual revolved around agile development and the development process. This was a deviation from the rest of the answers and was never specifically mentioned when asking other interviewees. None of the respondents who pointed out that they had a larger group project together with a company stated that they had any challenges regarding the development process during the start of their professional career. Testing is something which the majority of respondents did on a daily basis. However only one mentioned it as something they found challenging during the start of their career. Similarly, the one respondent who had not had experience working with a group project felt that working with, and understanding requirements were something that was challenging. One individual who primarily worked with User Experience design expressively mentioned that their skill regarding Computing Foundations was lacking.

The results of the analysis are displayed in the graph below.



*Figure 1. The percentages on the graph were assigned depending on the frequency of the topics being mentioned during the interviews*

There were areas where the mapping was not applicable and risked out on clouding the information which was given by the respondents. These areas are written below:

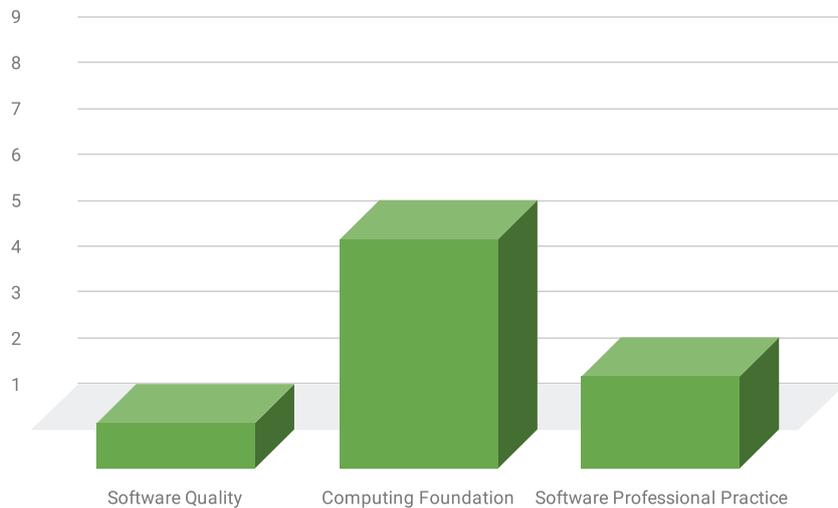
Several respondents said that there is a large difference between the real world and the school. On the other hand, one individual noted that one main part regarding this is that in the real world a project has a lifespan meanwhile during the education it's simply an assignment. The school offers a more safe and secure point of view, which was something the respondents felt was different in the real world.

Almost half of all people interviewed stated that they had difficulty getting an end-to-end understanding of the development process. Two of the interviewees pointed out that some newly graduated, including one of the interviewees, initially didn't understand that it is okay to make mistakes. It is common for newly graduated to put pressure on themselves while also believing that they can do anything. One respondent mentioned that during the start of their career, they felt that working with frameworks was challenging. Stating that the school was outdated when it came to these areas.

Only one person explicitly mentioned version control management as being a challenge. However, when other questions revolving version control was brought to light, almost all respondents said that they required initial learning on how to properly work with it during the start of their professional career. It was also highlighted that the education they took didn't provide any means of learning it apart from it being a requirement during their group project.

### 5.1.2 Strengths

In the questionnaire towards the newly graduated, there was a section which aimed at identifying what strengths they possessed and what the school made them good at. Seven of the areas touched upon could be mapped to the knowledge areas of SWEBOK. The greatest strength which five of the newly graduated believed they had was a good understanding and skill concerning areas related to Computing Foundations. Two of the respondents pointed out that their skill in communicating and cooperation was one of their greatest assets. One respondent mentioned that a strength which was gained from school was the ability to differentiate quality. Where the first solution you come up with isn't always the best. This was not similar in other cases where the respondents stated that the way they were reviewed and graded was simply by either being passed or fail.



*Figure 2. The graph displays the percentage and ranking of the topics mentioned as a strength during the interviews with newly graduated. 66% of the respondents mentioned that they felt confident when coming to new techniques, such as a new programming language and new frameworks. The reason for this confidence according to the respondents was because they had been exposed to several different techniques during their school life. One respondent mentioned that their flexibility came from not having become deeply engaged with a certain language, which made them more accustomed to familiarize themselves to new environments. In conjunction with this, two respondents explicitly mentioned that the school had given them a stable base to work from. This meaning that they felt the school gave them information on several different areas making them more comfortable around different SE topics.*

Two of the respondents mentioned that as a new graduate, one of their greatest strengths was that they possessed a desire to learn new things and improve. One respondent mentioned that as a new graduate you offer a different approach on how you look at things compared to more experienced developers. Insinuating that the respondent could provide a new way of thinking for future development.

### 5.1.3 Improvements

In the questionnaire towards the newly graduated, there was a section which aimed at identifying what they felt they wanted to improve on. The result data indicated that there are several areas which the newly graduated specifically wanted to improve upon. Almost 56% of the employees mentioned that Software Testing was something they wished to improve upon. Another 33% highlighted their interest to become better at software development. These individuals also said that they were not poor programmers, but had realized that they could always evolve and become improved developers. Two respondents commented that they wanted to emend their architectural skill and get better at Software Design.

Following that, one individual mentioned that maintenance was something which the respondent wanted to enhance. One of the newly graduated brought up that one thing that they wished to develop was their communication and cooperation skills to upgrade the team-work together with their colleagues.

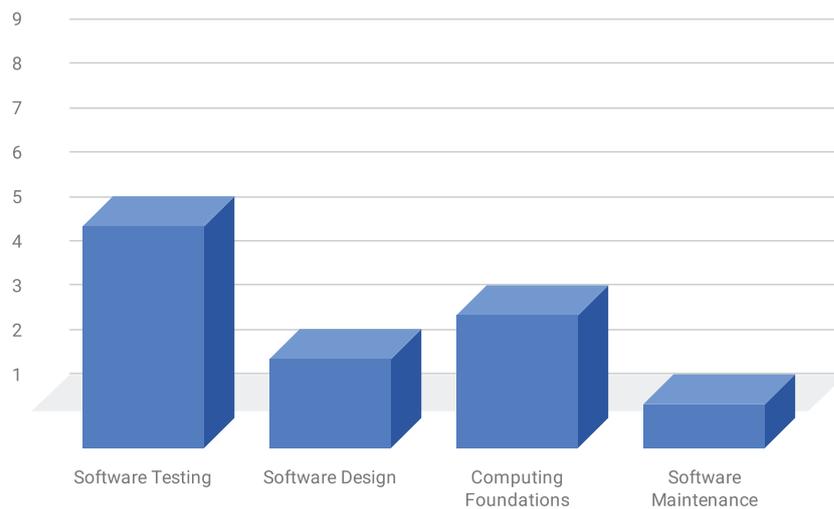


Figure 3. The graph displays the percentage and ranking of the topics mentioned as a property which the employees would like to improve on.

The end-to-end understanding of the working life was something that 33% named as something they wanted to strengthen. Two respondents mentioned techniques or frameworks which they wanted to become better at. One individual noted that understanding of and writing clean quality code was something which this individual wanted to improve upon.

## 5.2 Recruiter

### 5.2.1 What the recruiters wanted

Of all the 13 interviewees, four of them were recruiters. Among the recruiters, there was a clear pattern on what was considered important in a candidate.

All of the respondents mentioned that they considered soft skills to be one of the most important aspects when considering a new employee. Wanting to develop, both as a programmer and as a person, curiosity, cooperative, and be sympathetic; were some of the soft properties wanted. At the same time, all respondents also brought to light that it is important to have some programming foundation and a general interest in the area to stay relevant as a recruit.

### **5.2.2 What the students were struggling with**

One recruiter mentioned that students sometimes think that they are fully learned when they leave school, which is not the case. This respondent continues to mention that it is important to know that there is always room for improvement. Just because you have finished school, does not mean that you have learned everything. It is important to be humble, especially when facing new tasks, for example, taking the step from school to the industry. Another recruiter said that new recruits need to know that they are beginners and new to the business, it is okay for beginners to make mistakes.

But it is not just the soft skills that students are struggling with. Half of the recruiters bring up the subject of quality code. One mentions that school is doing a great job introducing areas such as requirements, estimating work, planning, and development. At the same time, school is lacking when it comes to the quality of a product. In student projects and laboratories are taken to a quality level where it works one time, but in the business, it is important to be as consistent and thorough as possible. Meaning that the product needs to be sure to work in as many situations as possible.

One important statement that one of the respondents said was different from the other recruiters' answers was that the respondent claimed that the school did a bad job on adapting their programming foundation for how it works in the industry. This recruiter claimed that the school was working with outdated techniques and was hindered by the national school policies to be relevant. The same respondent also said that the school had its focus wrong. In school, the students develop the best shell sort, but that is a task which almost none of the students will face during their career. Continued with the statement that the areas which they teaching are comfortable with is where the student also will go. Usually, the teachers are not enough up-to-date which is being displayed on some students.

## **5.3 Their concatenated opinions**

### **5.3.1 What about code?**

Some of the newly graduated commented that they felt that their programming foundation was one of their strengths when graduating. That may be the case, but some recruiters were not fond of how the newly graduated students were writing code. None of the recruiters said that the students are bad programmers, but instead that their code was not the best when it comes to code quality. Even though quality is a very broad notion, all descriptions come down to how the

code is written. The recruiters agreed that if the code is easy to read, it is a good code. Quality code may also refer to code that is easy to maintain. Though one recruiter mentioned that "quality code does not require any maintenance", it is still good if the code is easy to work with. Presumably, it has to do with the students' lack of experience. This was neither something the recruiters mentioned as a serious problem, but something that the newly graduated should be aware of. Even with this information at hand, is it possible for the school to inform the students about their coding capabilities?

### 5.3.2 Regarding end-to-end perception

Almost all recruiters stated that the newly graduated lacked an understanding of an overall picture. The process of receiving an assignment and turning it in is something that sticks with the newly graduated from their academic time. This was also shown as being one of the challenges that some of the newly graduated recognized they faced and wanted to improve upon. A group project together with a company is the closest encounter students have to these areas. Although both of the parties understand that this property is something that comes with experience, an action could be taken by the school or by the company to early on show the entire end-to-end process of a product. A clear difficulty in this is the nonlinear way a project is constructed. Different projects are done differently and the variety of techniques can vary enormously from project to project.

### 5.3.3 When it comes to SE Economics

Software Engineering economics was the second most challenging area according to the newly graduated. However, none of the recruiters mentioned that they experienced it as an issue which the newly graduated face. One newly graduated mentioned that there doesn't have to be something in depth regarding the topic, however, it is important to create an awareness of the financial structure and actual cost which goes into projects in the industry. Since these were areas which were barely touched in school, incorporating them in some way could clearly benefit the newly graduated's conception of the economy in the real world.

### 5.3.4 In the Professional Practice

Six of the respondents mentioned that they in some way found that areas related to Software Engineering Professional Practice were challenging. The themes which were mapped into the knowledge area were mainly areas revolving around cooperation and communication. One of the recruiter respondents said:

*"Their personality weighs more than their technical knowledge."*

Which is strengthened by another recruiter who mentioned:

*"It is easier to improve one's technical knowledge than their soft features."*

These two statements are further empowered by the other recruiters, which translates directly to a heavier emphasis on the soft characteristics of the employees. One important thing to take in consideration that was mentioned from

some of the recruiters is that it is not just the soft characteristics that determine how employable a worker is. Another mention from the recruiters was about the importance of balance. It is impossible to have a company with employees that all have strong soft characteristics. The same is also true the other way around, where the employees possess heavy technical knowledge but lack of soft properties. Therefore, it is important to make sure that all employees possess a shared knowledge library from both of these areas. But with this in consideration, the emphasis is still heavier on the soft characteristic, because of one the simple reason already mentioned. *It is easier to improve one's technical knowledge than their soft features.*

Another question from the questionnaire regarding the importance of soft skills also followed up if the school reflected the importance of soft skills of a software developer. Most of the respondents did not think that the school reflected that importance, but had at the same time difficulties coming up with a way for the school to represent this reality. Suggestions both from some respondents and from previous studies indicate an emphasis on group-oriented projects, preferably with real companies. Another method to put emphasis on soft skills in a school environment is to include study visits in the curriculum. To get the students out from the school environment and get an understanding of how it looks in the real world, might have positive results for students understanding regarding how you work with Software Engineering.

## CHAPTER 6

---

### Results

#### 6.1 Research Question 1

**Are there any software engineering areas where the recently graduated software engineering students frequently struggle when beginning their careers in the industry today?**

The areas which were stated by the newly graduated which could be mapped to the knowledge areas of SWEBOK are in order: SE Professional Practice(1), SE Economics(2), SE Maintenance(3), SE Testing, Computing Foundations, Software Models and Methods, Software Requirements coming next. The biggest challenges which could not be mapped are: Working Insight, End to end understanding, Comprehend that it is okay to make mistakes, Version Control, Modern Technologies.

This result highlights the fact that these are areas where the school needs to put more of their focus on. The education which was provided was not sufficient enough to provide the students with a good enough understanding of these areas. By looking at the analysis requires the most attention is the Software Engineering Professional Practice knowledge area. The currently most used way of providing real-life experience for the students is through a group project which some schools provide. Through the group project which some schools have the students are able to get a first-hand touch of the Software Engineering Professional Practice area.

#### 6.2 Research Question 2

**Do new employees experience other challenges than recruiters believe new graduates have? If yes, which areas differ?**

Even though there were just some employees who mentioned their struggle regarding professional practice, all the recruiters agreed that the newly graduated students mostly lacked soft skills. Based on the opinions of the recruiters, it is common that newly graduated students do consider themselves being fully educated, which is not the case. It is vital to understand the necessity of always being able to improve. Newly graduated are actually beginners on the working market, which some students do not realize.

There is also some struggle for the students when it comes to the technical aspects as well. The recruiters talked about quality when it comes to both a product and the code. In school, the students work towards making a working product, while in the industry, it is not just about getting the product to work, which half of the recruiters mentioned. This mindset is according to some of the

recruiters latched in and can become quite a challenging threshold to cross. A similar scenario applies to the code quality. Many students write different code, which makes the code personal. If the code is personal, it may be tricky to read. Self-explanatory code, which is impersonal, is our preferred way of writing code, according to one recruiter.

Even though the students have their shortcomings, they are still doing several things correctly. Almost all the recruiters experience that the students are overall average developers. They are familiar with agile development method, can solve problems, and have had some "real-world" experience in their group projects. Although there are some areas that need improvement, for example, more code maintenance, an up-to-date technique such as frameworks, and a larger emphasis on project and economy, the school is doing a rather good job at educating the students.

### 6.3 Research Question 3

**Based on the result from the two other research questions from this study, does a knowledge gap exist between the university and the industry? If yes, what actions can universities and companies take in order to address the gap?**

On the note regarding the potential gap from school to the industry, the answer is yes. There exists a gap, even though it may differ for separate persons. According to the recruiters, the school is teaching outdated techniques, not using frameworks, and that its code policies are outdated. The idea of this study was to find out if there was a gap between the school's education when compared to the industrial needs, and this is something confirmed from the results of this study. Even though some gaps can be individual, there are some joint gaps which is where the school could target their education at. The results gathered from research question one and two both indicate that there exists a gap, along with data from the analysis. This paper can be used both by the school and by companies hiring newly graduated students.

#### 6.3.1 How can the school use this study?

Further down, there is a section called *Suggestions to school* which contains recommendations which the school could use to improve the education towards Software Engineering students. The school can use the recommendations provided by this study, which are based on the combined views of the recruiters and the newly graduated students. All opinions come from the interviewed recent graduates who have mentioned what they felt they were missing when graduating and the interviewed recruiters. It is not the intention of the school to do everything for the students, as it still requires their own involvement from the students to learn, but the recommendations can be used in such a way to facilitate the process for the students to find out how the industry works. One main takeaway from this study is that there could be an emphasis put on understanding how money works in the industry. Students also need to get a better understanding on how everything functions when they start their career and

then they need to get a better end to end understanding of how projects are constructed and worked within the industry.

By educating students who better match the needs of the companies, students from the specific school also become more attractive in the working market. This, in turn, increases the school's reputation and directs to more people applying to that school.

### **6.3.2 What about the companies?**

The primary factor for companies to keep in mind after reading this study is that the school cannot provide everything for a student. The school's job is to give the students as wide an education as possible in the chosen field so that the students themselves can choose what they want to do afterward. No employer will require the exact same skill-set, and the school tries to cover the most important factors, but cannot facilitate every single requirement. One of the recruiters commented that it is not just the school's responsibility to educate the students. In accordance with this, the recruiter also mentioned that the business community almost has as much responsibility to ensure that employees can grow and continue to learn new things during their career. This is something that benefits both the worker and the employer.

Furthermore, one thing to mention is that it is difficult for the school to teach experience. Experience comes with practice, and practice comes from work. This study can be used by employers to get a deeper understanding of what the current newly graduated students may struggle with regarding their first job after graduating. With the fortified understanding, the employers can better accommodate the newly graduated students when they come to their first jobs, which in turn, makes for a better employee.

## CHAPTER 7

---

### Discussion

Several recruiters commented on students perception of themselves when they graduate, that students are believing themselves done with learning, which is a false statement. Some students have attended school their entire lives, so who is there to blame when the students think themselves finished when graduating? Because the students may not possess any working related experience, it is easier to see oneself as done, when there is nothing to compare against.

## 7.1 Comparison to other studies

The Systematic literature review made by Garousi et al. had several different sections which brought up the skills in the industry and their priority according to the result of their study [4].

### 7.1.1 Most important skills

One section brought up the most important skills in the industry. The result provided was compiled from all the studies in their review and showed that requirements, design, and testing were the most important and also most frequently-mentioned topics. The topics SE professional practice, project management, and development came next. However when looking at the most recent papers in their study the three most noted topics are SE professional practice, project management, and testing. For our comparison, the technical skills were extracted: requirements, design, testing, development. In the study, they explicitly mentioned that requirements, testing, and design were all more important than the actual development.

When asking the respondents about this, almost all agreed. There was a slight deviation regarding opinions on the top technical skills, where some individual considered that development would be higher up on that list. Two respondents brought up that the utmost important thing is to have the possibility to learn new things and possess modesty. These can be linked to the Software Professional Practice knowledge area. Strengthening the claim which the results ranging from (2013-2018) displayed as being the most important topic. When we asked our respondents about this list, including both the newly graduated employees, and the recruiters, they could give their opinion on how well this list matched their opinion and what they had experienced.

### 7.1.2 Least important skills

The least important skills mentioned in the systematic literature review ranging from the papers between (1995-2018) were: SE Economics, Mathematical Foundations, and Maintenance. The opinions shared by the newly graduated had a variety of different responses. One conception being that maintenance should

not be an area which is ranked so low. One respondent stated that usually maintenance is one of the things which you experience during the beginning of your career and is something which you will probably work with during your entire career. Another response was that almost 50% of all development is written for a maintenance purpose and that the two go together. Two other opinions shared were that Software Engineering Economics should be listed as the most important of the three. All respondents shared the opinion that Mathematical Foundations was the least important area and was something they directly hadn't used during their time as an employee.

When comparing the opinions shared to the papers which were published between (2013-2018) they match in the area regarding maintenance. This could highlight the fact that the focus of the industry regarding maintenance is shifting during later years. However, the opinions regarding Software Engineering Economics do not align with the result from the papers. Four out of nine newly graduated agrees that it needs to be brought up in school, while in the report it is deemed as the least important area. One similarity is the opinion of one recruiter which stated that Economics in Software Engineering is bad. In contrast, another recruiter stated that the areas are all in some way related to each other and are all equally important. The same study contained a section where they identified which areas contained the largest knowledge gap and how important they were.

### **7.1.3 Largest gap, and most important skill**

The first part of the result highlighted topics which were important and where the greatest knowledge gap existed. According to the study, the area where the greatest knowledge gap existed was Software Professional Practice. This is something which goes in line with what the results from the analysis that this study displayed. The topics which the newly graduated believed were the biggest challenges showed that six out of nine interviewees thought Software Professional Practice was something which they found difficult.

### **7.1.4 Largest gap, but least important**

The second part contained topics which were believed to contain the biggest gap but were the least important. These areas were Software Engineering Models and Methods and Maintenance. The opinions of interviewees showed that the majority did not agree that maintenance was not an important area. According to some of the respondents, a lot of their time is spent watching other people's code and can be a skill which is of benefit when trying to understand an already existing system. The interviewees did, however, agree that Software Engineering Models and Methods was not especially important.

### **7.1.5 Smallest gap and most important**

The result also covered topics which were considered to be important but contained the smallest gap. One topic was placed in this area: Development. The opinions of the interviewees stated that they agreed to the placing of the topic

in this area and highlighted the importance and how it functions as a base for further improvements as a Software Engineer.

### 7.1.6 Smallest gap and least important

Down at the final segment listed topics which contained the smallest gap and were the least important. The topic which fit into this category was Mathematical Foundations. The placing of the topic in this area was supported by all interviewees, however, it was brought up that in some areas mathematical skills are important, for example when working with a game creator och when working with machine learning.

### 7.1.7 Soft Skills

The systematic literature review also mentioned the importance of soft skills in the software industry. In the review, they pointed out that 24 out of the 33 studies recognized the importance of soft skills. A study performed in New Zealand by Watson et al. [16] reported that:

*"Soft skills are critical skills in SE and make up seven of the top eight most important skills[.]".*

Aasheim et al. made a study in 2009 [16] on the topic, which was cited in the SLR, that also recommended:

*"Soft skills and business skills must be included in curricula".*

This matches the result that we got from our analysis. All of the recruits specified the weight of soft skills, and that it is slightly more significant than the hard skills. Some of the newly graduated stated that the way which the school reflected this was through group projects where cooperation between other students was necessary to succeed. The opinion was not shared by all of the interviewees. There was some that stated that the school did not put any emphasis on this and that the results of the group projects were not always positive. Thus making the learning experience provided by the school quite sub-optimal.

## 7.2 How can the results be used

All the interviews have been conducted in Karlskrona, where the largest university is BTH, the majority of respondents have also attended BTH. Consecutively will lead to several comparisons to BTH and their education curriculum. This does not mean that this study is irrelevant to other schools, but consecutively, find out what BTH does right, and what needs improvements, so that the schools can then adopt the results of the study to their own.

### 7.2.1 What did the students like about their education?

Both employees and recruiters had opinions regarding the school's education. It was highly spoken of that the school's group project was one of the best things that the school did. The fact that the school had group projects together with

companies was what developed me the most, both as a person and as a developer, mentioned by one employee. There were many similar opinions regarding these group projects from both parties.

As stated in the Software Engineering curriculum [11]:

*”The core in education is the project courses where the students get the chance to apply their theoretical abilities in software development. These projects strive to resemble projects which the student will face in their upcoming line of work [..]. In the projects, the student is faced with an external client and requirements manager, which means that the student combines theoretical knowledge and generic skills.”*

Even though the citation has been translated, it still holds the same meaning. The above quotation fits in well with the description that both recruiters and employees gave to their reasoning why this was a good idea. It is as close to reality as possible, which is why it gave me the most out of my education, is a comment from another recently graduated student. A closer look at the curriculum shows that there are three unique courses with a project that follows this manner. When asked if one of the interviewees wanted to see more group projects in their education, there were mixed answers. Some brought up that the other courses are necessary for them to be able to have these project courses, and that there would be too many projects if more of the same would be added to the education. It might be troublesome for companies involved to force projects just for the sake of the course, is a shared opinion from one of the respondents. This claimant also said that it is better to have fewer project courses with a greater commitment from the companies side. Others discussed the possibility to either add more projects, but with a greater focus from the school’s side, where other courses try to follow similar concepts, without the need of industry involvement.

### 7.2.2 Improvements desired by the students

When asked if there was something the school needed to change, several respondents brought up direct opinions to this matter, while some had difficulties naming anything in particular. The compiled opinions of what school would need to add to their education would be:

- Version Control Management
- Better and more testing courses
- Architectural design
- Better user experience courses
- More cooperation
- Modern technologies
- More development

Next section will cover areas which the school can improve.

## 7.3 Suggestions to school

During the interviews, the newly graduated were asked to specify areas which they believed the school should focus more on. The areas derived from experiences they had gathered so far during their career as a Software developer. It is still important to remember that the school cannot do everything for the students, because the labor market is very saturated, and the school is not able to customize their education to a single line of work or profession. The result of the data showed:

### 7.3.1 Version control

Almost all respondent mentioned that they used a version control system, for example, Git, daily. It is a small but important part of all our respondents work, but nowhere is it mentioned anywhere in the Software Engineering curriculum [11]. None of the authors have experienced any version control system during their school time. It has been mentioned sometimes and we could use it if we wanted to, but there has never been any formal education on version control systems. Together with the data from the interviews, all respondents that had mentioned a version control system also agreed that it was something school could be teaching. It does not require to be a separate course but could be included from the very first programming course to embed version management in the course, so that all students had their own repository. Continuing courses could then also utilize this skill so that the students become more accustomed to working with version management than they are today. It does not require the school to represent a large company-like scenario with a big repository with several contributors to the code base, but just to accustom the students.

The purpose of the version control introduction is to inform the student of the current state which version control has in the industry. By awakening the idea to the students, it becomes easier for them to learn and understand the principle of version management themselves.

### 7.3.2 Testing

The majority of respondents mentioned that they felt that the skills they had acquired during their education regarding testing were inadequate. Testing is something which the majority of people we interviewed did on a daily basis. One interviewee specifically mentioned that an emphasis on the practical part on how to properly write tests was more important than the theoretical parts.

To improve upon testing in school, several respondents agreed that testing should be brought up earlier in education. "When I started programming in school I did not know anything about testing, was one comment from an interviewee. If the educators would include small testing parts into the development phase, the whole testing mindset would permeate through all the code that I would write."

Not all respondents mentioned this as a potential "fix" to the lack of testing in school, but it does not mean it is a bad thing to try. All the code the respondents wrote, did always have a test case to it. If the idea of testing would

arise earlier, it would be easier to facilitate the same principles held at different companies regarding testing.

### **7.3.3 Modern technologies and Frameworks**

The majority of interviewees worked with or had worked with some sort of framework. The result data indicated that although due to the difficulty in adapting frameworks into a curriculum the majority of newly graduated all worked with some sort of framework. An insight on how to work with frameworks is according to the result data something that could be of benefit to include in the school curriculum.

It is important to first know how to work with a programming language before adapting frameworks, but the school could add the application of frameworks to the education.

### **7.3.4 Real life experience**

Getting a grasp on how the real world operates and how to cooperate with people is something that was mentioned during almost all the interviews. Although the majority of interviewees had partially done a suiting task which matched the area during their time of studying. They described it as an experience which is very valuable to have before starting their career. The task which the interviewees had experienced was two group projects which matched the mentioned area. Mainly the last final one which was done during the last period of their education.

To get the students to touch reality by having group projects similar to the real world is one way to make the students get some experience.

### **7.3.5 Quality**

Some of the interviewees also believed that during the time they spent studying they rarely got critiqued for writing code which might not be considered to be optimal. They felt that the assignments that they were given simply had the criteria of either passing or failing. Whereas in the real world the code you write has to function properly for all possible scenarios.

By instead giving more feedback on the code and teaching a structured way of writing code would give an increase to the students programming.

## **7.4 The limitations of the study**

While this study wants to cover as big a scope as possible, it is not possible for it to cover everything. All of the interviewees were somehow involved in working at a consulting company. This limits the variety of different responses which were able to be gathered. The specific requirements for working at a consulting company inevitably differs from other professions in the Software Engineering industry. For example, a product company has a different focus compared to the consultants workday.

## 7.5 Validity Threats

Another important point to mention is that all of the interviews were translated from Swedish to English. This poses a risk of misinterpretation between the reader of this study and the actual meaning of the results. The opinions derived from the interviews can be interpreted as something very crucial, while in fact it was not that important at all. The same can be said the other way around, that a subject was loosely mentioned that it needed a small improvement, but actually required a substantial change.

All suggestions and responses in this study consist of interpretations from both the authors and the respondents from the interviews. They can not be considered a concrete fact. That does not mean that the information and results provided by this study are inapplicable, but instead that they need to be taken into consideration that it is based on individual opinions.

Almost all of the respondents have a degree from BTH, which can give an angled view from this specific school. The results from this study may not be the case for other schools or other locations for that matter. If there had been more time, and a larger quantity of people who were interviewed the results might have been different. To interview larger diversity of people could have generated another result.

## CHAPTER 8

---

### Conclusion

To conclude this interesting study, there are several areas which newly graduated software engineering students find challenging. By conducting interviews with nine newly graduated employees and four individuals who in some way are responsible for recruiting these areas were highlighted. The areas which are according to the newly graduated were notably most challenging and thus require the most attention by the school and their educational plan are:

- SE Professional Practice
- General picture
- SE Economics
- SE Maintenance
- End to end understanding

The other areas which were brought up were: As a beginner, it is okay to make mistakes, Version Control, Modern Technologies, Software Requirements Software Testing, Computing Foundations, and SE Models and Methods. All recruiters stated that the most important thing when hiring a new employee is their perception of the individual's soft skills. The soft skills heavily outweigh the technical skills in most areas and are typically harder to learn when comparing the two.

The area which most of the newly graduated students struggle with according to the recruiters are the soft skills, modesty is something which plenty of new employees lacked according to all of the recruiters. Almost all respondents mentioned that a large portion of their time is spent using tools such as Git, various testing tools and frameworks. This is something the school should take action upon and try to incorporate in a better way into their curriculum. Similarly, the industry should be aware that these are areas not so heavily touched in today's education. The group projects which some universities have with cooperation from companies in the industry is the closest thing which the students come in contact with that mirrors how working in the industry actually is. However, relying on one large group project to represent the topic which contains the largest challenge is quite treacherous as if something goes wrong the area fails to get addressed properly.

## REFERENCES

---

- [1] A. Begel and B. Simon, “Novice software developers, all over again,” p. 12, 2008. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/novice-software-developers-all-over-again/>
- [2] Det här är bth [this is bth]. Accessed at: 2019-05-28. [Online]. Available: <https://www.bth.se/om-bth/det-har-ar-bth/>
- [3] A. D. Radermacher, “Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectation of industry,” p. 117, 2012. [Online]. Available: <https://pdfs.semanticscholar.org/c6cc/c1a275c39010d1ea087b03d26e74752d9a3a.pdf>
- [4] V. Garousi, G. Giray, E. Tuzun, C. Catal, and M. Felderer, “Closing the gap between software engineering education and industrial needs,” *IEEE Software*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8664487>
- [5] P. Bourque and R. E. Fairley, *Guide to the Software Engineering Body of Knowledge (SWEBOK)*, 3rd ed., 2014.
- [6] S. Kvale, S. Brinkmann, and S.-E. Torhell, *Den kvalitativa forskningsintervjun [The qualitative research interview]*, 3rd ed. Lund, Sweden: Studentlitteratur, 2014.
- [7] M. Barmark and G. Djurfeldt, *Statistisk verktygslåda 0 - att förstå och förändra världen med siffror [Statistical Toolbox 0 - Understanding and Changing the World with Numbers]*, 1st ed. Studentlitteratur, 2015.
- [8] P. Aspers, *Etnografiska metoder : att förstå och förklara samtiden [Ethnographic methods: to understand and explain the present]*, 2nd ed. Malmö, Sweden: Liber, 2011.
- [9] “Good research practice,” Vetenskapsrådet, 2017. [Online]. Available: <https://www.vr.se/english/analysis-and-assignments/we-analyse-and-evaluate/all-publications/publications/2017-08-31-good-research-practice.html>
- [10] A. Begel and B. Simon, “Struggles of new college graduates in their first software development job,” *SIGCSE Bull.*, vol. 40, no. 1, pp. 226–230, Mar. 2008. [Online]. Available: <http://doi.acm.org.miman.bib.bth.se/10.1145/1352322.1352218>
- [11] “Utbildningsplan för software engineering [education plan for software engineering],” Blekinge Institute of Technology, 2018. [Online]. Available: [http://edu.bth.se/utbildningsplaner/PAGPT\\_HT-16.pdf](http://edu.bth.se/utbildningsplaner/PAGPT_HT-16.pdf)
- [12] T. C. Lethbridge, “The relevance of software education: A survey and some recommendations,” *Annals of Software Engineering*, vol. 6, no. 1, pp. 91–110, Mar 1998. [Online]. Available: <https://doi.org/10.1023/A:1018917700997>

- [13] C. Aasheim, J. Shropshire, L. Li, and C. Kadlec, “Knowledge and skill requirements for entry-level it workers: A longitudinal study,” *Journal of Information Systems Education*, vol. 23, no. 2, pp. 193–204. [Online]. Available: <http://jise.org/Volume23/n2/JISEv23n2p193.pdf>
- [14] J. Liebenberg, M. Huisman, and E. Mentz, “Knowledge and skills requirements for software developer students,” *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, vol. 8, no. 8, pp. 2604–2609, 2014. [Online]. Available: <https://pdfs.semanticscholar.org/48c8/b82b0837aea5fc618a39857eea0a14fb3579.pdf>
- [15] V. Clarke and V. Braun, “Thematic analysis,” *The Journal of Positive Psychology*, vol. 12, no. 3, pp. 297–298, 2017. [Online]. Available: <https://doi.org/10.1080/17439760.2016.1262613>
- [16] C. Watson, K. Blincoe *et al.*, “Attitudes towards software engineering education in the new zealand industry,” in *28th Annual Conference of the Australasian Association for Engineering Education (AAEE 2017)*. Australasian Association for Engineering Education, 2017, p. 785.