Identification of Skills to Improve Software Developer Productivity

Ramith Lasitha Wanniarachchi

179136H

Degree of Master of Business Administration in Information Technology

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2021

Identification of Skills to Improve Software Developer Productivity

Ramith Lasitha Wanniarachchi

179136H

The dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfilment of the requirement for the Degree of Master of Business Administration in Information Technology.

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2021

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

……………………………….

R. L. Wanniarachchi.

(Signature of the candidate) Date:

The above candidate has carried out research for the Masters thesis under my supervision.

……………………………….. ………………….

Dr. Shantha Fernando Date

Signature of the Supervisor

COPYRIGHT STATEMENT

ABSTRACT

Software Engineering has become one of the essential requirements for today's business, and software development has become one of the trending employments today. Due to the business's technical advancement and competition, the IT industry's expectation of software development skill level increases. Therefore, the industry is looking for more qualified software developers even though software developers are available. Some companies arrange technical improvement training or/and soft skill improvement training to improve software developers' skills. These training programs improve the software developer's skills to a level required at the moment, and some training programs are too late to initiate. As a lesson learned from that when hiring fresh graduates, some companies provide training before employment. Therefore, this research aims to find out the skill gap of the software developers in Sri Lanka and determine which skills impact the software developer's productivity, and find out how to improve the software developer's productivity. A survey was carried out among a sample of software developers. A series of interviews were conducted with industrial experts to determine the aspects influencing the software developer's productivity and present significant factors. Based on the analysis undertaken to the collected data, the results revalidate that the software developer has to strengthen soft skills such as Team Working, Problem-solving, Communication Skills, and improving programming skills to improve the software developer's productivity. The results revalidate that even though programming skills are considered the key skill for a software developer, the software developer should improve soft skills to benefit programming skills. Industrial experts suggest several approaches to improve these skills. Based on the survey result and expert feedback, sharing knowledge in platforms, taking online courses, and trying out new technologies can improve software developers' productivity by improving technical skills.

Further, the findings show that training such as leadership and team working training and business communication training should improve soft skills. This training was suggested to be arranged by the employer based on the business requirement. Employers should encourage the employees to do self-learning and try out new technologies. Education institutes should also provide training that focuses on soft skills such as teamwork and communication to improve software developer's soft skills. Also, expert feedback suggested that the education institutes should encourage students to take online courses to match the IT industry's expectations.

ACKNOWLEDGMENT

I want to expose my sincere thankfulness to all those who supported me in completing my research study on "Improving software developer's productivity." Initially, I would like to thank my thesis supervisor, Dr. Shantha Fernando, for the guidance and support provided throughout the research study. He was always available there to clear my doubts and questions about the research or writing. He constantly allowed this to be my work while guiding me in the correct direction. Also, I would like to thank Dr. Dilum Bandara, the former coordinator of the MBA in IT, for the encouragement and guidance and all the other faculty members in the CSE department at the University of Moratuwa.

I would also take this opportunity to thank the industry experts who took part in interviews for this research project and the support I got from the software developers who participate in the online surveys to provide their valuable input. The following research dissertation was carried out with the help of the people known and unknown to me. Finally, I appreciate those who helped me in many ways, although the names which are not mentioned above but were useful for the success of this study.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

List of Abbreviations

AI – Artificial Intelligence

ANOVA - Analysis of variance

IT – Information technology

PMI – Project Management Institute

PMP - Project Management Professional

Q4 – Fourth quarter of the year

ROI – Return of Investment

SDLC – Software Development Life Cycle

UML - Unified Modeling Language

US – United States

1. INTRODUCTION

1.1. Background

Information technology has become the most growing industry globally due to the importance of implementing information technology in all the world's businesses. Some business outsources IT solution into other IT companies, and some companies maintain an IT department inside the business. Therefore, Software development has become one of the most important roles for all the business. Since ROI from IT cannot be measured directly, companies must consider the historical financial facts to find out the ROI from the IT solutions. Therefore, Software developer's productivity has become a significant concern nowadays to deliver the best quality product to the business.

Software development companies can be divided into two major categories, project-based software development companies, and product-based software development companies. For software developers in project-based companies, they may have to work on different projects based on the project that comes into the company. Therefore, the expectation from the company from the developer is changing based on the project allocated. Software developers working in product-based companies may not get much difference in company expectations. However, expectations may change with time. Due to the company's expectations, software developer's productivity has become everyone's attention these days. Some companies rate software developers based on their performance and give additional benefits to motivate them to work productively. Some companies track the software developer's development activities through a dashboard and use other tools to measure code-related factors. How many software developers focus on identifying the skill gap and providing guidance or training to improve their productivity is a concern.

1.2. Motivation

Software developers are educated from different institutions with different training backgrounds, and the education level of the software developers is different. Therefore, the software developer's productivity depends on several aspects, and there could be common aspects as well. Most of the companies provide Internships before starting to work as software developers. However, some complained that the Interns primarily work on the background work, such as writing unit tests to achieve the unit test coverage. Some are getting better exposure to the appropriate software development activities. Some companies provide training before starting working as a software developer to ensure the employee can work as a software developer in the IT industry. Still, companies are trying to focus on the deliverable without considering the software developer's abilities to work on the software development project.

1.3. Problem Statement and Research Question

The problem statement addressed in this research is that:

Due to Technical and Soft skill barriers that exist with most software developers, the productivity of most software developers is not as expected by the industry to fulfill their task effectively.

Due to Technical and Soft skill barriers that exist with most software developers, most software developers' productivity is not expected by the industry to effectively fulfill their task. To improve the software developer's productivity, the software developer must find out the skills gap that exists with the industry and improve those areas to improve productivity. Therefore, the author developed the following research question:

What approaches can be used to improve software developer productivity?

1.4. Research Objectives

This research aims to determine how software developers can improve their skills and improve productivity to meet industrial expectations. Therefore, the researcher will determine what skills impact the software developers' productivity to meet the expectation as the first step. The skill gap between the industry expectation and the software developer's skill level needs to be identified to determine the necessary skills to improve the software developer's productivity. Based on the skill gap, the skills affecting software developer's productivity are identified. After identifying the skills affecting the developer's productivity, the final objective is to find out how to improve the software developer's productivity based on the software developers' feedback and the experts' feedback. It then provides recommendations to improve software developer's productivity and when those actions need to be taken. Therefore, the objectives of the research can be listed as follow:

- Identify the skill gap between the industrial expectation and the software developer's skills
- Among the above-identified skills, the next step is to identify what skills need to be improved to improve the productivity of the software developer
- Find out how to improve the productivity of the software developer

1.5. Research Scope

As discussed in the above topics improve software developer's productivity has become an essential fact in the IT industry. Therefore, it is necessary to identify the skills affecting the software developer's productivity and address those skill gaps to improve its productivity. Identifying the skill gap and find out how to address those skill gaps to improve the software developer's productivity is beneficial to the IT industry. The scope of this research is to,

- Refer to related studies to identify the research background and the skills affecting the software developer's productivity.
- Gather inputs from software developers working in the IT industry to find out the skill level and skill expectation of the industry and then calculate the skill gap compare to the expectation of the IT industry
- Analyze feedback provided by the software developer to identify the available approaches to improve the software developer's productivity
- Gather expert feedback from the interviews conducted for Industrial experts related to improving software developer's productivity.
- Propose how to improve the software developer's productivity based on the facts identify the above steps.

1.6. Research Significance

Significance of this research to find the approaches to improving the software developer's productivity and next most significance to find out the skills affecting improving software developer's productivity. Based on the research outcome, software developers can focus on each aspect and approach to enhance their productivity give out the best to the IT industry. Apart from that, IT companies can optimize their internship. Training and guidance can be provided to the software developer to maximize productivity to maximize ROI from the IT solutions applied to their businesses. As the software developers benefit, they will complete their tasks on time and maintain the deliverable's high quality.

1.7. Outline

The rest of the chapters are organized as follows, chapter 2 describes the literature review on improving software developer's productivity, and it covers, Importance of productivity, Required knowledge and skills for software developers, Absence of software development skills, Improving software developers skills, and Academic contribution to improving software developer's skills topics. Chapter 3 discusses the methodology used for the research and describes the research method, population, sample selection, and data collection method. Chapter 4 discuss the data analysis process, and it covers demographic, inferential, descriptive, and regression analyses and presents the expert feedback collected. Chapter 5 presents the observations, recommendations, and conclusion based on the data analysis, and this chapter provides the research limitation and future work as well

2. LITERATURE REVIEW

2.1. Introduction

Information technology has become vital to business nowadays. IT facilitated the company's smooth operation and provided solutions to improve ROI (Quinonez, 2014). With the technical advancement, companies are involving IT to expand their business, and software engineering has become vital in today's industry. The software developer is playing an essential role in making software solutions for the company. Therefore, the productivity of the software developer has come to attention. This chapter describes the software developer's productivity and how it can improve productivity to complete work effectively.

2.2. Importance of software developer's productivity

Employees are considered the company's assets and, the business needs to improve the productivity of the employees to improve the ROI of the business (Wagner, 2018). Primarily, an individual's productivity in a company is measured by comparing the work done with the employee's average, while the employee's perceived productivity can be measured by looking from the employee's perspective. To measure the Perceived Productivity, the software developer can self analyze the skills and the improvement can they can make to work complete more work (Meyer, 2017) effectively. Based on the analysis conducted in the same article, they have observed that perceived productivity is a highly personal matter where the perception is different across developers. Except for the skills of the software developer, there are other factors affecting the software developer's perceived productivity, such as planned meetings and work unrelated browsing as less productive activities (Meyer, 2017).

Effectiveness and productivity have differences where effectiveness is the effort the individual puts in, and productivity is the effort's outcome (Wagner, 2018). When a company hires an individual, the company invests in developing the business, and if the employee productivity is impacting the return. Therefore, productivity is one of the essential facts when considering the return on investment. The other fact is the deliverable quality, where most of the world's business is now integrating with IT solutions in their business. The IT solution needs to have expected quality and fewer issues that should not interrupt the business and productivity (Mtsweni, 2016). Therefore, when improving the software developer's productivity, the deliverable quality also needs to maintain at an expected level.

2.3. Required knowledge and skills for software developers

Compared to other professions, software engineering is one of the most knowledge-intensive professions. Required education attributes for software engineering include more technical skills as well more on soft skills. Software developers develop software, and they require a set of skills to build the software. Science, math, engineering, and design techniques are essential to software developers when developing software solutions (Matturro, 2013). This article suggests that soft skills such as teamwork, multitasking, communication, analytical, critical thinking, troubleshooting, creativity, and technical skills are essential for a software developer. Based on the study conducted on (Aasheim, 2009), the software industry expects soft skills such as team working, analytical skills, creative thinking, and interpersonal skills from a software developer to work productively. Therefore, the following topics will discuss those skills in detail.

2.3.1. Team-working skills

Unlikely other professions, teamwork plays a significant role in software development since several developers carry software development. In the end, the product is a combination of those components developed by the software developers. During software development, every developer must ensure that everyone on the same page avoids misunderstandings and delivers the product before the deadline(Matturro, 2013). Software development is teamwork where the software development team is delivering the product. A software developer may have to help other software developers since every developer is not an expert in all the technologies and programming languages. It is all about grooming up the team by teaching others and documenting new things for others, so the team's productivity matters end of the day (Li, 2015).

2.3.2. Time and task management skills

As explained above, teamwork is essential for a developer to deliver a product as a team. To achieve team goals time and task management plays a significant role where some developer depends on other developer's work and estimation need to be done for work the work to identify the timeline. So the team productivity depends on each developer's time and task management (Li, 2015). It is not completing work faster, and it is about completing the task without affecting the quality in less time. Proper planning and task prioritization will lead to completing a project as planned (Bailey, 2014). Therefore, time and task management skill is an essential skill for a software developer to work productively.

2.3.3. Communication skills

Communication plays an essential role in the IT industry since the IT employees should align with the client's requirements. Communication involves several areas, such as identifying the requirement that the client wants, clarify the requirement if there are dark areas, provide new ideas, and give the current status to the client. If the IT employee cannot perform these tasks, the client would see the IT employees not doing the work that is expected, even the IT employee productively working on client requirements. Communication management knowledge is another necessary factor for successful project delivery. Auditing, communication, competencies, conflict resolution, environment, intention, language, status reporting, technology, time, and trust relates to communication management.

Most importantly, trust between IT team members and every stakeholder can be improved by effective communication. Communication skill plays a critical role in sharing thoughts and understanding others' opinions. Teamwork communication plays a significant role in making sure there is no misunderstanding between developers and every developer understands what needs to be done to deliver the product (Matturro, 2013).

Considering software failures, most of the software failed due to a communication gap. Software developers have to focus on improving communication skills since they need to spend some going through the requirement and understanding what needs to be implemented. Communication skills play a significant role when communicating and sharing the software developer's thoughts because the developer has to develop what business required (Mtsweni, 2016). Therefore, communication skills are considered as one of the key soft skills for a software developer.

2.3.4. Interpersonal skills

As mentioned above, communication is one of the primary skills required for a software developer, and interpersonal skills always develop communication skills. Interpersonal skills help software developers interact with each other more comfortable way and understand each other's thoughts. This skill allows a developer to dive into the career further because software development is more involved in teamwork (Ahmed, 2013). Interpersonal skill is not learned during the classroom, and it needs to learn at a tender age, and it becomes a part of the personal behavior (Bailey, 2014). This article points out that the lack of interpersonal skills is one reason for poor carrier path software development.

2.3.5. Problem-solving

Once the design has taken place and when it comes to the implementation, problems are like to arise, and the software developer has to have the ability to solve the problem. Software developers spend more time debugging rather than writing new code, and finding out the issue is not sufficient. A software developer needs to have problem-solving skills to resolve the problems observed during software development. Problem-solving skills are considered essential skills required for a software developer (Morales, 2011).

Software developers are expected to have good problem-solving skills since the work is done in the IT industry is something new, and different types of problems are expected to observe. It employee required other skills such as reuse of code to save their time developing the same principle, training, and staff development is needed to improve the skills of the IT staff and maintenance, and support is required to maintain the project which the IT staff has developed

2.3.6. Analytical Skills

To solve the problem, the software developer must identify the issue, and analytical skills are considered to analyze details. It helps to identify the issue and the root cause. Analytical skill helps everyday activities such as identifying patterns, interpreting data, and making decisions (Morales, 2011). This skill helps the developer gather the correct information, solve complex issues, and effectively execute it (Bailey, 2014).

2.3.7. Creative thinking skills

Creativity is about how the problems are solving effectively in software development. A creative solution is not the quickest solution or the most complex solution. It is about finding out the best solution for the problem (Graziotin, 2014). This solution could be specific to the situation, which may not be the best solution to another similar problem. When the software developer finds the most optimal solution to the problem, the software developer must improve the solution to solve it more optimally, which is called more creative (Graziotin, 2014). Being creative as a software developer provides several advantages, such as saving time and effort, reusability, and quality of the deliverable.

2.3.8. Programming skills

Programming skills are the key skill for a software developer. Some software developers learn several languages and technologies, and most developers focus on a few programming languages and are not focusing on other languages much. Learning all the relevant programming languages even at a basic level, the software developer can perform well during employment than developers who know few programming languages (Morales, 2011).

A software developer's technical skills can be defined as the abilities and knowledge needed to perform specific tasks. While technical skills are often most significant for software jobs and other jobs related to science, technical skills are common in other industries also. Most of the technical skills require training and experience to improve the skills and are considered hard skills. Hard skills are the skills that can be learned, defined, evaluated, and measured. Therefore, IT students are initially supposed to focus on technical skills and soft skills in current education since technical skills are mandatory for the IT industry.

2.4. Absences of software development skills

Based on the survey conducted in (Litecky, 2010), 25% of responses highlight that it is hard to meet the job listings' skills. Therefore, the problem is that software developers do not have a skill level up to the IT industry's expectations. Compared to other job streams, Blockchain Developer has the highest skill gap compared to other streams, and .Net developer has a better skill level than required by the industry.

Comparing only the software development streams (Litecky, 2010) has highlighted that the Java and .Net developer has more skills than required by the industry. Other development steams need improvement to match the industrial expectation. Even though Java and .Net developer has the employer's skill, most employers are not seeking Java and .Net developers. There are looking for software developers on other streams with less skill than expectation (Litecky, 2010). Therefore, this article suggests that the software developer has to improve the skills based on the job's expectations and not the industry requirement. The software developer has to learn skills based on a job interested in instead of basic programming languages.

Another article (Rajlich, 2013) points out that 50% of employees lack technical skills underemployed. It points out that not keeping the skill update to date with technology advancement is the main reason for the lack of technical skills. Furthermore, mobile and cloud solutions have more involvement in today's business, and software developers have to focus on learning mobile and cloud technologies rather than desktop technologies (Rajlich, 2013).

According to (Edwards, 2004) problematic features to the software engineering profession are:

- The tension between systems development and maintenance/support work

- A combination of organizational and technical aspects
- The nature of team working
- A combination of generic skills and extremely specific skills
- Constant change, some of it externally imposed
- The need for a quick response coupled with long system lifetimes

When considering software engineers' work, a software engineer has involved with software development and software maintenance tasks, which can be characterized into two parts such as creative intensive exciting part and the boring, routine, annoying part. Based on the article's details, very few software engineers are involved with only technical work or organizational or managerial responsibilities. Table 1 (Edwards, 2004) shows a broad characterization of the relationship between these responsibilities and the earlier activities. This balance, or indeed tension, between technical and organizational activities is an issue to which we shall return later.

Table 1: Responsibilities of Software Engineering Activities

Software engineering Activity	Main Responsibilities
Investigation	Organizational
Determine Feasibility	Organizational
Systems Analysis	Organizational, Technical
System Design	Technical, Organizational
Programming	Technical, Managerial
Testing	Technical, Managerial, Organizational
Training	Organizational, Managerial
Documentation	Technical, Managerial
Implementation	Organizational, Technical, Managerial
Maintenance/Support	Technical, Organizational, Managerial
Project Management and Control	Organizational, Managerial
People Management	Managerial

This problem is not unique to the software engineering profession, and some of the other professions are also suffering from this issue. It implies the importance of knowledge management to the IT industry. Employees can improve organizational knowledge by providing proper knowledge during higher studies or providing specific knowledge before the IT organization's work (Edwards, 2004).

Article (Colomo-Palacios, 2010) points out that most computer science students focus on Operating systems and barely new technologies such as AI, DevOps, and Cybersecurity. Therefore, higher education is not aligned with the expectation of the fast-growing industry. The same article points out that the students' soft skills are also not aligned with the industry expectation. When those students become software developers, software developers' productivity seems very low compared to other expert software developers.

Article (Surakka, 2007) points out that there are several causes for software developer's skill gap, and the main three reasons are listed below,

- The need for IT is spreading
- Supply is decreasing due to an increase in demand for software engineering
- The way of learning software engineering is changing

The need for IT is spreading – Not like traditional software development, nowadays the requirement is different. Software developers have to more involved in cloud computing and artificial intelligence, and they have to use work with new trends such as SaaS services rather than building the solution in house

Supply is decreasing due to an increase in demand for software engineering. With the evolution of technology and the business requirement, business requires more software developers to maintain the existing system and build new functionalities to the IT solution. The skills necessary for the job are changing. Therefore, considering the graduates released from universities, the supply is not meeting the business's demand (Surakka, 2007).

Learning software engineering is changing – Article (Surakka, 2007) points out that people learn software development through programming boot camps that focus on hard programing skills and missing out on a formal degree from a university or college. Therefore, the companies miss out on these talented software developers since they don't have a university degree. Furthermore, these software developers miss out on soft skills that require working as a software developer within a collaborative environment.

2.5. Improving software developers' skills

As discussed in (Borstler, 2001) article, there is little agreement about what a software engineering project course should cover. Some project courses cover detailed aspects, e.g., programming, usability and security issues, analysis, architecture, design, or work products. Provide software engineering project courses with real clients, large teams, and challenging problems is advised to improve students' knowledge and skills. Although it is hard to teach in a real client environment, they would enhance students' skills in various forms, and it will create software engineers as expected by the IT industry. Making students exposed to real client environments will improve their technical skills and non-technical skills, and they will also prepare to challenge them in their future careers in the industry. The article (Sedelmaier, 2014) describes that the software developer must master the fundamental skills first to be a great coder. Mastering the fundamentals of technologies used during the development will help the developer use technology with good knowledge and save time. The article (Knobelsdorf, 2008) describes that the software developer's mindset should be set with Curiosity, Team working, Pattern-finding, Skepticism, Fresh Perspective, Learning from failures, Communication, and Innovation. This article also points out that software developers have to master the technologies' fundamentals to improve their productivity. And it points out that the software developer should not learn all the programming languages. However, having a solid foundation in those languages will help the software developer work on new technologies. Reading code written by great programmers will allow the software developer to think in a new way when they are writing the code (Knobelsdorf, 2008).

According to (Mtsweni, 2016), the study has identified that effective communication can improve employee productivity and work quality. Improving the software developer's productivity should not affect the quality of employee's work. In (Li, 2015) study proposes technical and soft skills which could lead a programmer to be a great programmer. Among those skills, the study has identified that time and task management, deep and broad technical experience, and high-end user focus are the primary technical skills required for being a great programmer among other programmers and as soft skills. Further, this study has identified that a positive attitude, supreme communication skills, team playing skills, and quick learning ability would lead the programmer to be a great programmer. It proposes that graduation from a university with a degree would help the software developer identify the architecture better. The programmer must focus on improving the technical skills by following current trends, participating in blogs and articles, trying out new implementations, participating in conferences, and adding value to the community to improve its technical skills. It suggests that the programmers need to participate in conferences such as toastmaster's clubs and reading books will improve the programmer's required soft skills.

A study (Ahmed, 2012) found that software developers' demanding skills combine technical and soft skills by analyzing the latest job postings of over 700,000 jobs in the last few years. It has been identified that the demand has gone up in the last few years for the software developer's skills, and the software developers must improve their current skills to meet the expectations of the current market. Therefore, if the software developers have mastered the technical skills and lack soft skills, it would be a waste of technical skills since the job market's expectation is a combination of technical and soft skills. Considering the studies mentioned above, to improve the software developer's productivity, the software developer should consider improving their technical and soft skills to improve productivity.

2.5.1. Academic contribution to improving software developer's skills

Today there is a recognized necessity of teaching realistic software development in project courses. There are comprehensive courses, ranging from a single semester to two semesters courses, from a single client to multi-customer courses, from local to globally distributed courses, and from toy projects to projects with real clients. The challenge for a project course, which is not trivial enough, is to make the project complex sufficient to improve the students' software engineering knowledge and practice if it has a teaching environment that does not burden the instructor or the student's education process.

Even today, there is little agreement as to what a software engineering project course should cover (Borstler, 2001). Most academic programs have at least one such course, and a few offer several courses on the subject. Some project courses cover detailed aspects, e.g., programming, usability and security issues, analysis, architecture, design, or work products. The approach to simplify the project is to work in a problem domain that the students are even now familiar with or are interested in learning more about, such as computer game development or enhancing tools oriented toward program development. Then it will ease the teaching process to instructors, and students will learn with more enthusiasm. However, this approach's problem is that students often have a broad knowledge of the game or tool interface. Yet, they are unaware and inexperienced in the process of modeling, implementation, testing, and delivery. Constructing, understanding, and using models is an essential element of any software engineering course. Before using UML models for analysis and design, students used informal models when interacting with a client during early requirements engineering activities. Examples of informal models are film trailers to express the basic idea of the project to external stakeholders.

Although it is hard to teach software engineering project courses with real clients, large teams, and challenging problems, it will advance students' skills in numerous ways. They improve their technical skills and non-technical skills and prepare students with real challenges they will face in their future careers in the industry.

Employees working in the IT industry have developed their technology level by following a degree or an external program. Therefore, they build their mandatory technical knowledge by following a program. However, there are other skills that they need to develop which will not be covered by the degree of the external program. These skills can improve using different approaches such as reading books, blogs, and latest trends, participating in conferences, add value to the community by writing (Li, 2015), and attending to groups like toastmasters (Mtsweni, 2016).

Various experiments were conducted researching the capabilities of improving a programmer's productivity using existing programming languages and software development processes. For example, an experiment was conducted using two groups (Pavlov, 2017), and the first group of experiments focused on using only UML. As the communication within a team during an SDLC, they could not utilize natural languages such as English and Russian. The other group was allowed to use any language during SDLC. Both groups were assigned to do some tasks. They found that the first group permitted only UML as the communication had better productivity than the other group, which had no communication limitation.

In the ACM IEE software engineering curriculum, there are some factors highlighted about the education as well. This Curriculum points out that the Software Engineering education should be taught to set the mindset of the SD to work on large projects by practicing reading and understanding small projects to large projects. It points out that some topics need to be covered when the students get maturity with software engineering, and some materials should be taught at the beginning of the software engineering learning cycle, such as practical materials. Therefore, the students can gain maturity by taking real-world projects such as Internships, Open source projects, and other student projects. Another fact that point out in this publication is that students need to learn one or more application domain in-depth to apply to solve the problem later during software development. One of the other exciting points described in this publication is that they suggest that software engineering problem solving should teach in multiple dimensions. The student needs to develop their skills by solving problems such as analysis, design, and testing related to solving the client's issues. Teaching with up-to-date tools and using appropriate hardware is necessary to train the software developer to work in real projects and choose open source tools considered suitable due to the demand for software development. Since software development is rapidly evolving, this publication points out that reviewing and updating the curricula is one of the most important factors when teaching software development.

2.6. Summary

Since software solutions have been vital to the current business, teaching software engineering targeting the IT industry with relevant technologies and improving the relevant skills required at the industrial level is necessary. Based on the related work, there is a skill gap between the IT industry and software developers' required skills and knowledge, affecting their productivity. Therefore, related studies pointed out that there several skills that the software developer has to improve to improve productivity. Those skills are a combination of technical and soft skills since software development is teamwork, and it requires many soft skills such as communication and team working skills. Several approaches are highlighted in those studies to improve software developer's skills, and even the programming skill is considered the software developer's key skill. The software developer has to improve programming skills with the evolution of new technologies to improve productivity.

3. RESEARCH METHODOLOGY

This chapter will discuss the information related to the methodology for this research improvement of software developer productivity. This section included the sample size, sampling method, target population, scale of measurement, operational measurement, data analysis techniques used to analyze the gathered data.

3.1. Research Method

Section the Following figure illustrates the reach approach followed in this study. This approach contains steps taken, whereas the first step, the literature survey, was carried out to identify the skills affecting the software developer's productivity. Then based on those skills, the independent and the dependent variables have been identified. By going through the variables specified, the questions for the questioner been identified. While the questioner was shared with the audience, an interview for the technical experts was carried out to find expert feedback on improving software developer's productivity. After completing the survey, the data analysis has been done to identify how to improve the software developer's productivity by collecting data from the survey and data collected by interviews.

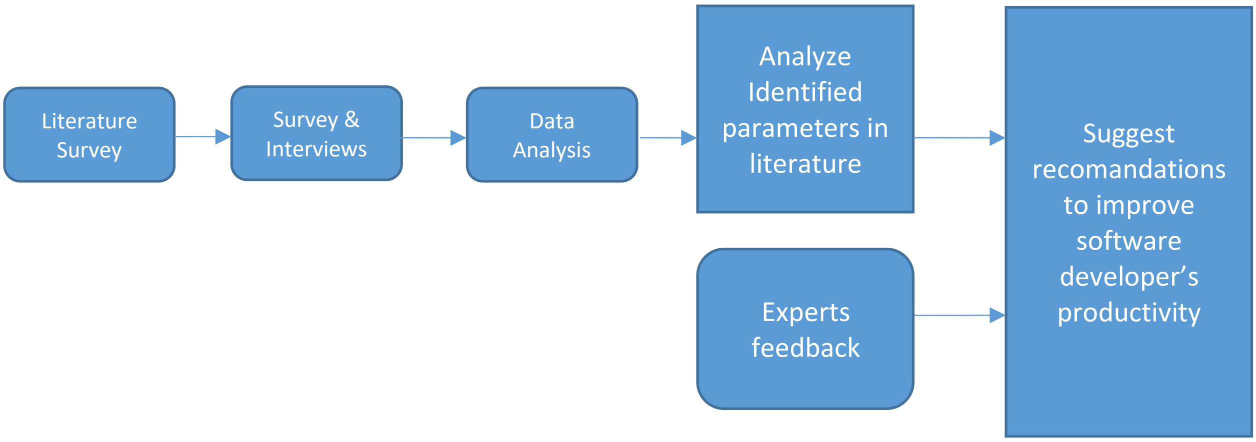

Figure 1: Research Methodology diagram

<u>Conceptual Framework</u>

Based on the literature survey, the skills that affect the improvement of software developer's productivity are displayed in the second column of Table 2. Regarding the sources, Independent and dependent variables are also identified fin column 3, where the independent variables are fixed, and Dependent variables are changed to measure the changes of independent variables. Out of those skills identified considered work, communication is recognized as one of the critical skills in most sources. Most of the skills identified are related soft skills of the developer.

Table 2: Factor table

Paper Title	Related Work	Skills Considered	Independent/Dependent Variable
Is There Still a Room For Programmers' Productivity Improvement?, 2016	Improving programmer's productivity	• Written communication skills	Level of written communication competency/Programmer's productivity
Knowledge and Skill Requirements for Entry-Level Information Technology Workers: Do Employers in the IT Industry View These Differently than Employers in Other Industries?, 2009	Entry-level knowledge and skill requirement and which skills specific to Software developers	• Systems development skills • Programming skills • Analytical skills • Creative thinking skills • Interpersonal skills • Communication skills • Team working skills	Technical and Soft Skill level/Programer's productivity
Soft Skills for Software Project Team Members, 2016	Improving software developer's soft skills	• Verbal Communication skills • Written Communication skills • Team Working skills • Creative thinking skills • Interpersonal skills • Time and task management skills • Problem-solving skills	Soft skill level/Programmer's productivity
What Makes a Great Software Engineer, 2016	Improving software developer's skills	• Team Working Skills • Creative thinking skills • Programing skills	Soft skill level/ Programmer's Productivity
Soft skills requirements in software development jobs: a cross-cultural empirical study, 2012	Improving software developer's soft skills	• Verbal Communication skills • Written Communication skills • Team Working skills • Analytical skills • Creative thinking skills • Interpersonal skills • Time and task management skills • Problem-solving skills • Programing skills	Technical and soft skill level/Programmer's demand
Important Job Skills for Software Engineers, 2019	Essential skills for software developers	• Written communication skills • Verbal Communication skills • Problem-solving skills	Soft skill level/Programmer's demand

Table 2: Factor table (Continued from previous page)

Managing Software Engineers and their Knowledge, 2014	Improving Software developer's skills and knowledge	• Communication skills • Problem-solving Skills • Programming skills	Technical and soft skill level /Software engineer's productivity
Non-Technical Skills for Success in a Technical World, 2014	Improving software developer's soft skills	• Verbal Communication skills • Written Communication skills • Team Working skills • Interpersonal skills • Problem-solving skills • Organizational skills	Soft skill level/Programmer's demand
Happy software developers solve problems better: Psychological measurements in empirical software engineering, 2014	Improving software developer's skills	• Verbal Communication skills • Written Communication skills • Analytical skills • Creative thinking skills • Problem-solving skills	Soft skill level/ Software Engineer's productivity
Analysis of Technical Skills in Job Advertisements Targeted at Software Developers, 2005	Improving Software developer's programing skills	• Programming skills	Technical skill level/Programmer's demand
Teaching developer skills in the first software engineering course, 2013	Teaching software development skills for entry-level developers	• Verbal Communication skills • Written Communication skills • Programing skills	Technical and soft skill level/Programmer's productivity
Identifying Technical Competences of IT Professionals, 2010	Software developer's programing skills	• Programming skills	Technical skill level/Programmer's productivity
Software engineering body of skills (SWEBOS), 2014	Improving software developer's skills	• Verbal Communication skills • Written Communication skills • Creative thinking skills • Programming skills	Technical and soft skill level/Programmer's productivity
Creativity as a pathway to computer science, 2008	Software developer's creativity	• Creative thinking skills • Problem-solving skills • Programing skills	Technical and soft skill level/Programmer's productivity

Based on the factor table, the research has identified nine independent variables by going through literature sources.

1. Verbal Communication skills
2. Written Communication skills
3. Team Working skills
4. Analytical skills
5. Creative thinking skills
6. Interpersonal skills
7. Time and task management skills
8. Problem-solving skills
9. Programming skills

The dependent value is identified as the software developer's productivity concerning the above-mentioned independent variables. This dependent variable is selected based on the objective of the research.

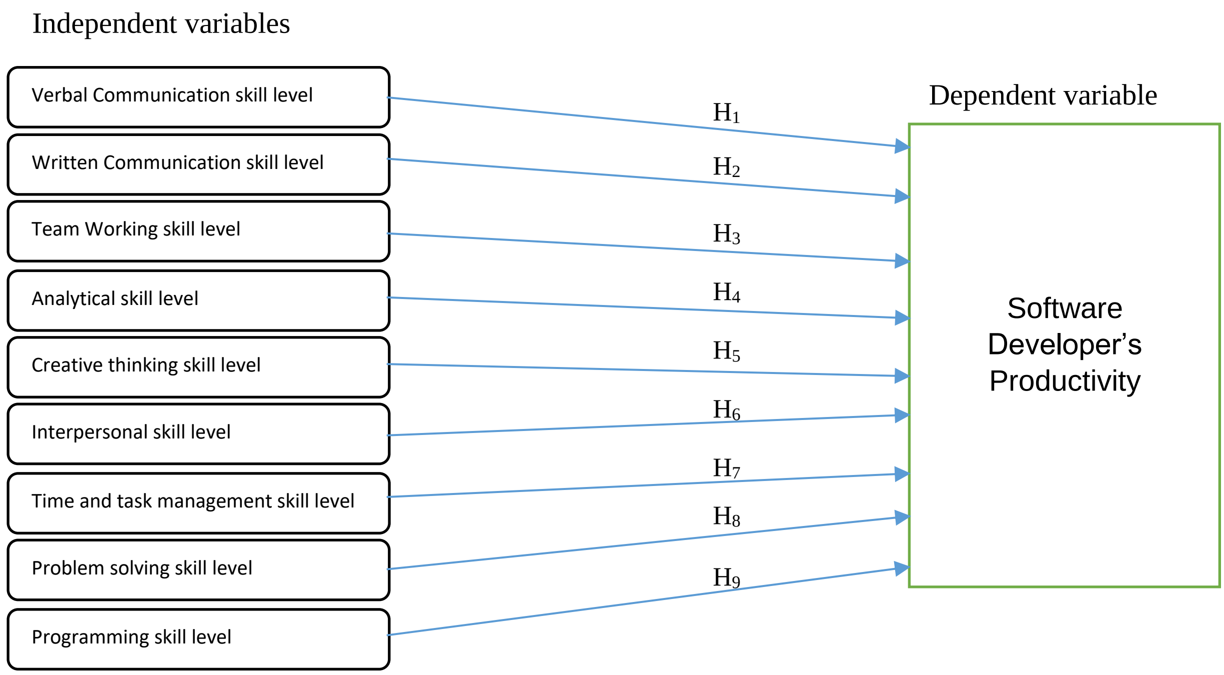

Figure 2: Hypothesis Mapping Diagram

As mentioned in the above figure, nine hypotheses are identified based on the literature reviewed, which affects the software developer's productivity.

Let;

HN_0: Null Hypothesis

HN_A: Alternative Hypothesis

H1

- $H1_0$ - There is no significant relationship between the improvement of Verbal Communication skill level and Software Developer's Productivity.
- H_{1A} - There is a significant relationship between the improvement of Verbal Communication skill level and Software Developer's Productivity.

H2

- $H2_0$ - There is no significant relationship between the improvement of Written Communication skill level and Software Developer's Productivity.
- $H2_A$ - There is a significant relationship between the improvement of Written Communication skill level and Software Developer's Productivity.

H3

- $H3_0$ - There is no significant relationship between the improvement of Team Working skill level and Software Developer's Productivity.
- $H3_A$ - There is a significant relationship between the improvement of Team Working skill level and Software Developer's Productivity.

H4

- $H4_0$ - There is no significant relationship between the improvement of Analytical skill level and Software Developer's Productivity.
- $H4_A$ - There is a significant relationship between the improvement of Analytical skill level and Software Developer's Productivity.

H5

- $H5_0$ - There is no significant relationship between the improvement of Creative thinking skill level and Software Developer's Productivity.
- $H5_A$ - There is a significant relationship between the improvement of Creative thinking skill level and Software Developer's Productivity.

H6

- $H6_0$ - There is no significant relationship between the improvement of Interpersonal skill level and Software Developer's Productivity.
- $H6_A$ - There is a significant relationship between the improvement of Interpersonal skill level and Software Developer's Productivity.

H7

- $H7_0$ - There is no significant relationship between the improvement of Time and task management skill level and Software Developer's Productivity.
- $H7_A$ - There is a significant relationship between the improvement of Time and task management skill level and Software Developer's Productivity.

H8

- $H8_0$ - There is no significant relationship between the improvement of Problem-solving skill level and Software Developer's Productivity.
- $H8_A$ - There is a significant relationship between the improvement of Problem-solving skill level and Software Developer's Productivity

H9

- $H9_0$ - There is no significant relationship between the improvement of Programming skill level and Software Developer's Productivity
- $H9_A$ - There is a significant relationship between the improvement of Programming skill level and Software Developer's Productivity

3.2. Data Collection

The data collection process is one important part of the research. The research findings are built based on the data collected from the research, and if the data collection were not done properly, it would drive the research into invalid results. Therefore, when collecting data, the researcher needs to understand the correct audience and organize it to get valid inputs from the survey or the interviews.

In this research, the author identified the audients from different companies and different levels of experience. The specific questions were asked to form the audients to get the correct inputs. Both qualitative and quantitative methods are used in this research to collect the relevant data. Quantitative data represent measurable data such as numbers and statistics, and qualitative data are more descriptive than quantitative data. Qualitative data cannot be measure directly like quantitative data, and it can collect through observations. The qualitative and quantitative data collected from the survey and the interview will be analyzed using different data analysis techniques, as discussed below. Based on the analyzed data, the observations will be presented.

3.3. Population and Sample Selection

This research aims to identify the skills that affected the Software developer's productivity and suggest improving the software developer's productivity. The survey was conducted by sending the questionnaire among software developers who worked in Sri Lankan IT companies. Most companies work on outsourced projects from international business and local business work when considering Sri Lankan IT companies. Most of the projects are international businesses. It covers most of the domains such as Healthcare, Banking, the Food industry, etc. When sharing the questionnaire among software developers, it was distributed fairly without focusing on one domain. Therefore, the audience of the questionnaire is from different domains and different technical divisions as well.

A survey was carried out among software developers in Sri Lanka to identify the skills affecting software developers' productivity. The target population was calculated based on the government statistics, <http://www.statistics.gov.lk/samplesurvey/2017Q4report.pdf, page 15> in 2017 Q4 74056 Software Developers were working in Sri Lanka. Therefore, the sample size to distribute the survey was 382, with a confidence level of 95% and a confidence interval of 5.

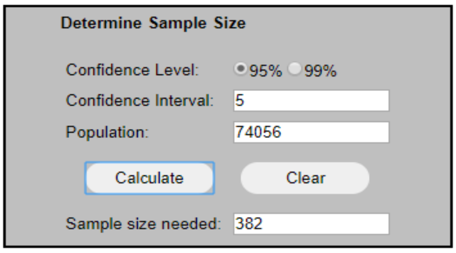

Figure 3: Sample Size

Along with the survey, a set of interviews are carried out, focusing on Managers and Team leads to get their opinion on improving the software developer's productivity. There is a set of predefined questions to identify those skills. There is a set of questions to get their ideas based on their experience working with software developers to improve software developer's productivity.

3.4. Process of Data Collection

Data were collected using interviews and sharing an online survey to collect the required data for this research. Interviews were conducted with industry experts to provide their expert feedback improving software developer's productivity. An online survey was shared with software developers in Sri Lanka to provide their input on the survey questions. The author has shared this survey through the Linkedin network by identifying software developers randomly in Sri Lanka.

The Questioner was created by adding 19 questions and dividing those questions into four sections. Section 1 addresses the demographics questions, and sections 2 to section 4 contain questions related to identified dependent variables. The below table displays the association of those variables with each section.

Table 3: Question Breakdown Table

Construct	Sub skills	Questions
Skill gap identification questions	• Verbal communication skill level • Written communication skill level • Team working skill level • Analytical skill level • Creative thinking skill level • Interpersonal skill level • Time and task management skill level • Problem-solving skill level • Programming skill level	2.1 to 2.9
Identify which education/training should improve	• Training provided during the higher education • Training gained by the internship program • Training provided by the company	3.1 to 3.3
Identify Other Factors	• Other factors that could improve productivity	4.1 to 4.3

Section 2 and 3 includes the answers gathered from the software developers using five Likert scales as follows,

Table 4: Answer Scale Table

Answer	Score
Strongly Disagree	1
Disagree	2
Some What Agree	3
Agree	4
Strongly Agree	5

Sections 4 contains questions that filter out the next question to ask the proper question based on the answers provided in the first level of the questions in section 4. Based on the responders' input on the first level, second-level questions appear as open-ended questions. The second level of

questions focuses on providing their opinion on improving software developer's productivity and suggesting a program or approach that can be used to improve software developer's productivity.

3.5. Summary

In this chapter, the author has discussed the research methodology used to collect data from interviews and the survey. This chapter describes the target audience, the sample size, and how it was selected for this study. This chapter describes how the reach methodology collects data from the survey/interviews from the research approach section. Then it describes how the observations are made from the data collected. Then it describes the skills identified from the literature review and identified independent and dependent variables from the skills. Based on those variables identified, the author identifies nine hypotheses, which will be evaluated through this research to identify the skills affecting software developer's productivity. Finally, this chapter describes that the research will point out approaches to improve software developer's productivity based on the research observations.

4. DATA ANALYSIS

This chapter discusses the data analysis for the data collected for the research from the selected target population. Descriptive and regression analysis was conducted during this data analysis to understand the relationship’s data better to improve the software developer's productivity. Additionally, an analysis was conducted on the training and programs available for software developers to find out the impact of improving productivity.

4.1. Data Preparation for Analysis

During the data collection phase, 458 responses were collected from 2019 August to 2020 January by sending the survey created to software developers working in Sri Lanka. The survey was shared online to collect data. In contrast, no missing data recorded.76 responses were removed from the collection since those responders were not software developers such as support engineers and system engineers by analyzing question 1, asking about the responder’s job description. The survey’s target sample size was to get 382 responses, while valid 387 responses were collected end of the data collection.

4.2. Demographic Analysis

Following demographic analysis, consider the analysis criteria such as Gender ratio, age groups, development experience, and development stream. During the survey distribution, this survey was shared across the community without considering the above-mentioned demographic criteria. Furthermore, this analysis simplifies the data analysis on the large number of data collected from the survey.

Considering all the valid responses collected from the survey, according to figure 4, out of 382 responders, 55 responders were female, and the gender ratio is calculated as 6: 1. When considering the Development stream, more male developers exist than female developers, but the survey was shared across the developers without considering the gender of the responder.

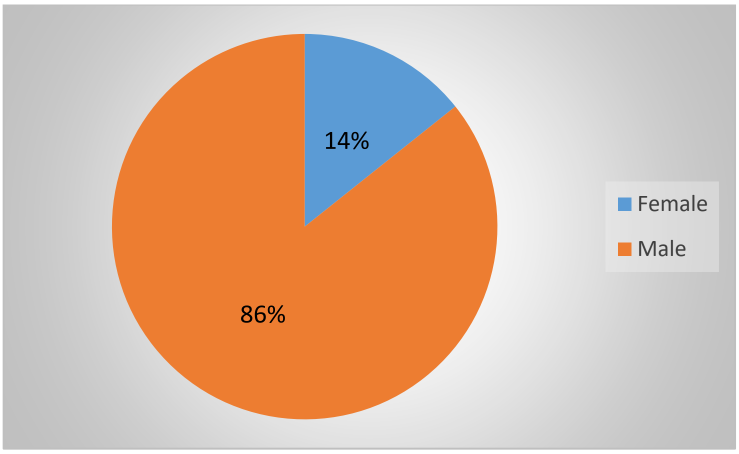

Figure 4: Gender distribution of the sample

During the survey, there was a question asked from the responders to mention the age group of the responder and based on the analysis, most of the responders are under age 26-30 age group as 184 (Figure 5) developers, and the next most responders are under 20-25 Age group. When considering the development stream, most of the fresh graduates(engineers) and senior developers (Senior Engineers) work under the supervision of the leads are can be considered in the 20-30 age group.

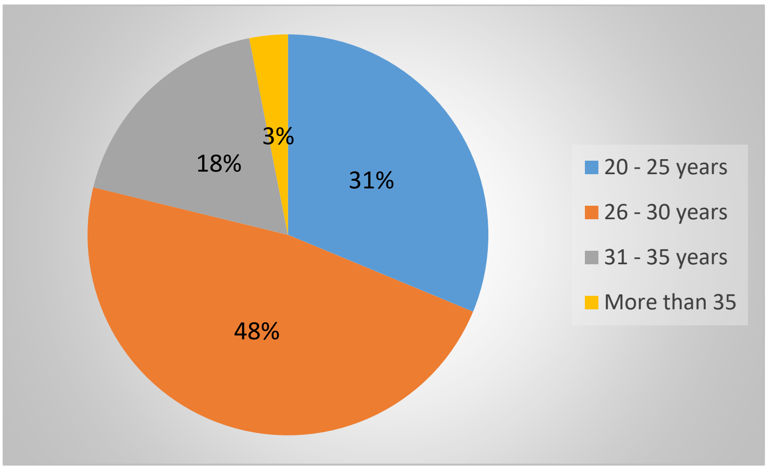

Figure 5: Age distribution of the sample

Even though the survey question about age, the responders asked another question to mention the years of experience on the development stream. According to figure 6, most of the responders have 2 – 5 years of experience (187 responders), and the next most groups are 0 – 1 year of experience (101 responders) and 6-10 years of experience(75 responders).

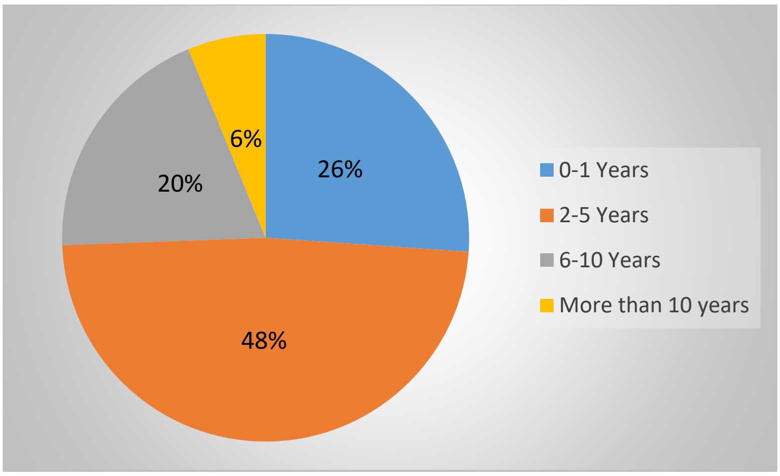

Figure 6: Experience distribution of the sample

In the software development industry, there are many technologies used for software development. Based on those technologies, most developers are used to stick one stream at their development task. Based on the collected data, most developers are used to do their development tasks with Java, and the second most are .Net. Figure 7 categorized all responses based on the technology, and Java developers are closed to half of the population (53% of total), where .Net developers are less than a quarter (15% of total) of the population.

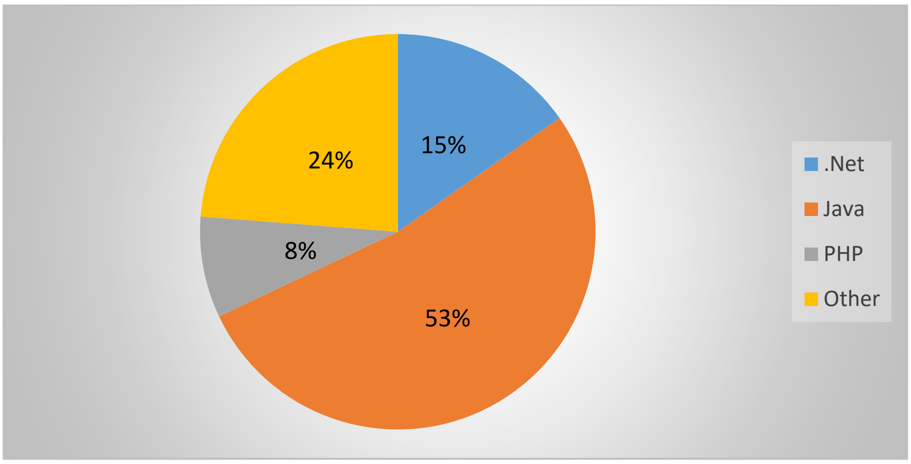

Figure 7: Technology distribution of the sample

4.3. Inferential analysis

As the next step, Inferential analysis has been conducted on categorized the data to identify the relationship between the variables and behaviors among the variables. Therefore, this analysis was useful to conclude after analyzing data against the methodology section's hypothesis.

The following table contains the mean of the developer's skill level and the skill gap comparing the expectation and the skill level for each skill considered in the hypothesis.

4.3.1. Descriptive and Regression Analysis

Before analyzing the data set, reliability analysis was conducted to validate the data set's consistency. Calculated results for reliability analysis are mentioned in the below table (Table 5). Based on the results, it is clear that the data set is high internal consistency since Cronbach's alpha is greater than 0.7.

Table 5: Reliability Analysis

Case Processing Summary

		N	%
Cases	Valid	387	100.0
	Excluded[a]	0	.0
	Total	387	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.943	18

As the next step, a descriptive and regression analysis has been conducted for skill gaps for each skill and productivity improvement. The calculated details are mentioned in the below table (Table 6).

Table 6: Descriptive and Regression Analysis

Skill	**Skill Level of the developer (1 to 5 scale)**	**Skill gap with the expectation (Expectation - Skills) (-1 to 3 scale)**	**Will Productivity improve filling the gap? (-1 to 3 scale)**	**Correlation "Skill gap" vs "Will Productivity improves filling the gap?"**
Verbal Communication	3.801	1.509	2.039	0.357 Sig(2-tailed) = .000
Written Communication	3.86	1.475	1.84	0.330 Sig(2-tailed) = .000
Team Working	**4.367**	**1.928**	**2.204**	**0.5** Sig(2-tailed) = .000

Table 6: Descriptive and Regression Analysis (Continued from previous page)

Analytical	**4.041**	1.69	2.132	0.408 Sig(2-tailed) = .000
Creative Thinking	3.935	1.61	2.075	0.394 Sig(2-tailed) = .000
Interpersonal	3.946	1.563	1.93	**0.454** Sig(2-tailed) = .000
Time and task management skills	3.848	1.656	2.202	0.431 Sig(2-tailed) = .000
Problem Solving	**4.119**	1.778	2.191	**0.447** Sig(2-tailed) = .000
Programming	**4.052**	1.724	2.165	0.492 Sig(2-tailed) = .000

According to the collected data, after calculating the mean value for Skill Level (0 to 5 scale) for each skill type, all the skills types rated more than three, which is more than the scale's median. And considering skill which has mean more than 4, responders believe that they have good skills in "Team Working," "Analytical," "Problem Solving," and "Programming." "Verbal Communication," "Written Communication," and "Time and task management skills" are rated with lover levels where the skill ratings are 3.801, 3.86, and 3.848 (Table 6) according to the sequence. According to Table 6, the "Team Working" skill type has the highest skill level, which is 4.367 out of 5.

According to skill gap analysis for data collected, there is a positive skill gap for all the skills considered in this research since all the values calculated are greater than 1 for each skill (-1 to 3 scale). Therefore, it was identified that there are skill gaps for all the skill types mentioned above based on the data provided by software developers. According to Table 6, "Team Working" has the highest skill gap value (1.93) and other skills that have similar skill gap levels vary from 1.47 to 1.78.

The next analysis was to identify software developers' feedback on will improving those skills up to the expectation to improve the software developer's productivity and considering the data provided. According to Table 6, it was identified that all the skill types have values that vary from

1.8 to 2.2 ~ 2 (2 => Agree in -1 to 3 scale). This stats describes that the Software Engineers believe that all the skill types will improve productivity by improving each skill level. “Team Working” skill type has the highest value of 2.204 (Table 6) for improving productivity by improving the skill compared to other skills considered.

After conducting the analysis mentioned above, correlation analysis was conducted among “Skip Gap” and “Will it Improve the productivity” to identify the relationship between productivity by filling the software developer’s skill gap. Based on the correlation calculated in Table 6, all the correlation values were positive and significantly were .000 for all the skill types. Which means there is a relationship between the skill gap and the productivity improvement for all the skill. However, the relationship level is different for each skill type. According to Table 6, “Team Working” has the highest correlation of 0.5, where written communication has the lowest correlation level of 0.330. Apart from that, “Interpersonal” and “Problem Solving” skills have the next highest correlation as 0.454 and 0.447.

One of the interesting findings was that the Team Working skill type has a higher skill gap of 1.928, the highest level of skills as 4.367, the highest value of productivity improvement as 2.204, and the highest correlation level as 0.5 based on the calculations in Table 6.

4.3.2. Hypothesis Testing

Hypothesis testing has been conducted to analyze the methodology chapter’s hypothesis based on the correlation analysis. Therefore, each hypothesis’s acceptance and rejection are mentioned in the “Hypothesis test summary” table.

H1

- $H1_0$ - There is **no** significant relationship between the improvement of Verbal Communication skill level and Software Developer’s Productivity.
- $H1_A$ - There is a significant relationship between the improvement of Verbal Communication skill level and Software Developer’s Productivity.

Test – According to correlation details provided in Table 6, verbal communication skill has .000 significant value and 0.357 correlation for improving software developer’s productivity by improving verbal communication skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.357 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_2

- $H2_0$ - There is no significant relationship between the improvement of Written Communication skill level and Software Developer's Productivity.
- $H2_A$ - There is a significant relationship between the improvement of Written Communication skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, written communication skill has .000 significant value and 0.330 correlation for improving software developer's productivity by improving written communication skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.330 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_3

- $H3_0$ - There is no significant relationship between the improvement of Team Working skill level and Software Developer's Productivity.
- $H3_A$ - There is a significant relationship between the improvement of Team Working skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, team working skill has a .000 significant value and 0.5 correlation for improving software developer's productivity by improving team working skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.5 (High correlation). Team working skill has 0.5 correlation with improving software developer's productivity, and it can be considered high correlation since the correlation is equal to 0.5.

H_4

- $H4_0$ - There is no significant relationship between the improvement of Analytical skill level and Software Developer's Productivity.
- $H4_A$ - There is a significant relationship between the improvement of Analytical skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, analytical skill has a .000 significant value and 0.408 correlation for improving software developer's productivity by improving analytical skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.408 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_5

- $H5_0$ - There is no significant relationship between the improvement of Creative thinking skill level and Software Developer's Productivity.
- $H5_A$ - There is a significant relationship between the improvement of Creative thinking skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, creative thinking skill has .000 significant value and 0.394 correlation for improving software developer's productivity by improving creative thinking skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.394 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_6

- $H6_0$ - There is no significant relationship between the improvement of Interpersonal skill level and Software Developer's Productivity.
- $H6_A$ - There is a significant relationship between the improvement of Interpersonal skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, interpersonal skill has a .000 significant value and 0.454 correlation for improving software developer's productivity by improving interpersonal skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.454 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_7

- $H7_0$ - There is no significant relationship between the improvement of Time and task management skill level and Software Developer's Productivity.
- $H7_A$ - There is a significant relationship between the improvement of Time and task management skill level and Software Developer's Productivity.

Test – According to correlation details provided in Table 6, time and task management skill type has a .000 significant value and 0.431 correlation for improving software developer's productivity by improving time and task management skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.431 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5

H_8

- $H8_0$ - There is no significant relationship between the improvement of Problem-solving skill level and Software Developer's Productivity.
- $H8_A$ - There is a significant relationship between the improvement of Problem-solving skill level and Software Developer's Productivity

Test – According to correlation details provided in Table 6, problem-solving skill has .000 significant value and 0.447 correlation for improving software developer's productivity by improving problem-solving skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.447 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5.

H_9

- $H9_0$ - There is no significant relationship between the improvement of Programming skill level and Software Developer's Productivity
- $H9_A$ - There is a significant relationship between the improvement of Programming skill level and Software Developer's Productivity

Test – According to correlation details provided in Table 6, programming skill has a .000 significant value and 0.492 correlation for improving software developer's productivity by improving programming skills. Therefore, the null hypothesis has been rejected based on the significant value, which is lesser than 0.05, and accepted the alternative hypothesis based on the positive correlation, which is 0.492 (medium correlation). However, the level of relationship is not strong since the correlation is less than 0.5.

Table 7: Hypothesis test summary

Hypothesis	**Null Hypothesis**	**Alternative Hypothesis**	**Level of the Relationship**	**Direction**
H1 - There is a significant relationship between the improvement of Verbal Communication skill level and Software Developer's Productivity.	Rejected	Accepted	Medium	Positive
H2 - There is a significant relationship between the improvement of Written Communication skill level and Software Developer's Productivity.	Rejected	Accepted	Medium	Positive
H3 - There is a significant relationship between the improvement of Team Working skill level and Software Developer's Productivity.	Rejected	Accepted	**High**	Positive
H4 - There is a significant relationship between the improvement of Analytical skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive
H5 - There is a significant relationship between the improvement of Creative thinking skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive
H6 - There is a significant relationship between the improvement of Interpersonal skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive
H7 - There is a significant relationship between the improvement of Time and task management skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive
H8 - There is a significant relationship between the improvement of Problem-solving skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive

Table 7: Hypothesis test summary (Continued from the previous page)

H9 - There is a significant relationship between the improvement of Programming skill level and Software Developer's Productivity	Rejected	Accepted	Medium	Positive

According to hypothesis analysis conducted above Table 7, all the alternative hypothesis has accepted, and all the null hypothesis has rejected based on the correlation scale which mentioned below Figure 8.

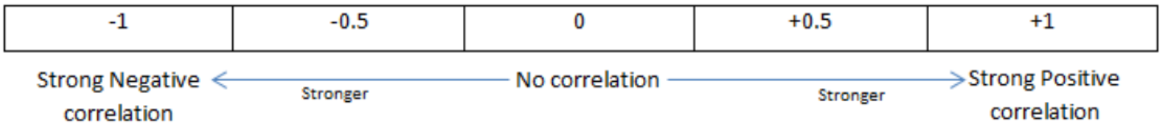

Figure 8: Correlation Statistics

But only the team working skills meet the Strong correlation, and other skill types have a positive correlation, and it is positive and equal to 0.5. Therefore, only team working skills have been marked as high for the relationship level.

4.3.3. Simple Regression Analysis

After conducting the correlation analysis for all the independent and dependent variables, all the independent variables are selected to further analysis for regression analysis to understand the relationship deeper than it was analyzed in the previous analysis. Therefore, the following analysis is conducted for each independent variable against the dependent variable to analyze more details on the relationship.

<u>Verbal Communication</u>

Table 8 describes model summary details related to productivity improvement (dependent variable) against verbal communication (independent variable). Based on the Adjusted R Square value, which is 0.125, 12.5% of the variance in productivity improvement can be explained by the skill gap of verbal communication. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 8: Model Summary for verbal communication skill gap

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.357[a]	.128	.125	1.037

a. Predictors: (Constant), SkillGapVerbalCommunication

Considering the ANOVA analysis in Table 9, the verbal communication skill gap calculated F value is 56.280, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 9: ANOVA analysis for verbal communication skill gap

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	60.506	1	60.506	56.280	.000[b]
	Residual	413.913	385	1.075		
	Total	474.419	386			

a. Dependent Variable: ProductivityImprovementVerbalCommunication

b. Predictors: (Constant), SkillGapVerbalCommunication

The following values were calculated for coefficients considering the verbal communication skill gap with productivity improvement. Based on the values calculated in Table 10, the distribution of productivity improvement with the verbal communication skill gap is plotted in the below graph (Figure 9). In addition to that, the R^2 line and fit line Y = 1.427 + 0.405 X was also defined based on Unstandardized Coefficients B values

(Y = Productivity Improvement and X = Skill gap)

Table 10: Coefficients for Verbal Communication

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.427	.097		14.709	.000
	SkillGapVerbalCommunication	.405	.054	.357	7.502	.000

a. Dependent Variable: ProductivityImprovementVerbalCommunication

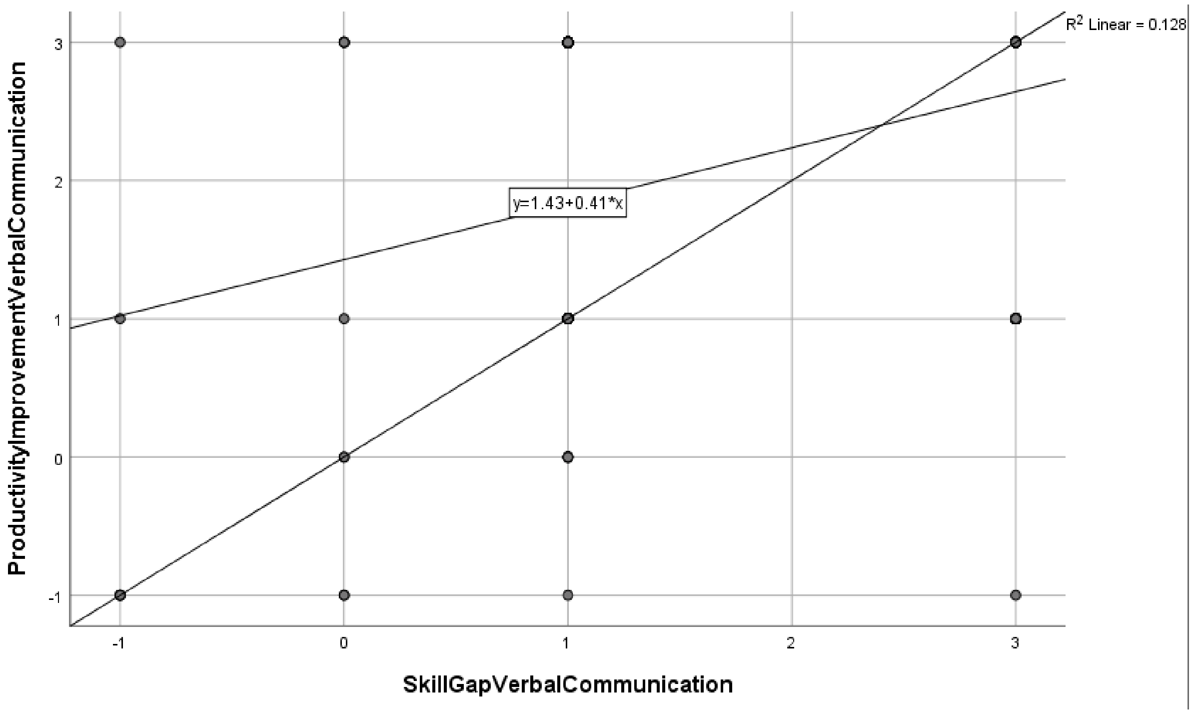

Figure 9: Verbal Communication vs Productivity Graph

<u>Written Communication</u>

Table 11 describes model summary details related to productivity improvement (dependent variable) against written communication (independent variable). Based on the Adjusted R Square value, which is 0.107, 10.7% of the productivity improvement variance can be explained by the skill gap of written communication. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 11: Written Communication Model Summary

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.330[a]	.109	.107	1.065

a. Predictors: (Constant), SkillGapWrittenCommunication

Considering the ANOVA analysis in Table 12, the written communication skill gap calculated F value is 47.126, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 12: ANOVA analysis for Written Communication

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	53.444	1	53.444	47.126	.000[b]
	Residual	436.623	385	1.134		
	Total	490.067	386			

a. Dependent Variable: ProductivityImprovementWrittenCommunication

b. Predictors: (Constant), SkillGapWrittenCommunication

The following values were calculated for coefficients considering the written communication skill gap with productivity improvement. Based on the values calculated in Table 13, the distribution of productivity improvement with the written communication skill gap is plotted in the below graph (Figure 10). In addition to that, the R2 line and fit line Y = 1.262 + 0.392 X was also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 13: Coefficient analysis for Written Communication

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.262	.100		12.611	.000
	SkillGapWrittenCommunication	.392	.057	.330	6.865	.000

a. Dependent Variable: ProductivityImprovementWrittenCommunication

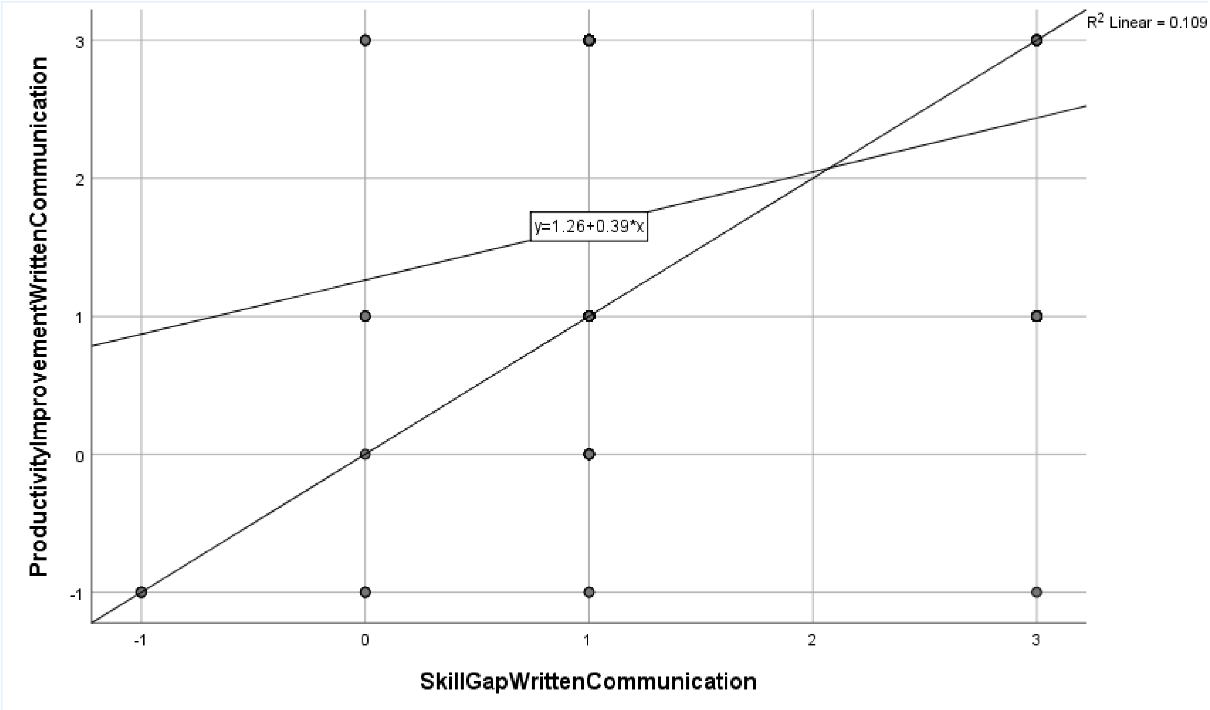

Figure 10: Written Communication vs Productivity Graph

Team Working

Table 14 describes model summary details related to productivity improvement (dependent variable) against the team working skill gap (independent variable). Based on the Adjusted R Square value, which is 0.248, there is a 24.8% productivity improvement variance explained by the team working skill gap. However, the relationship level can be considered a high relationship since the R-value is equal to 0.5.

Table 14: Model Summary for Team Working

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.500[a]	.250	.248	.972

a. Predictors: (Constant), SkillGapTeamWorking

Considering the ANOVA analysis in Table 15, the team working skill gap calculated F value is 128.407, and Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 15: ANOVA Analysis for Team Working

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	121.270	1	121.270	128.407	.000[b]
	Residual	363.603	385	.944		
	Total	484.873	386			

a. Dependent Variable: ProductivityImprovementTeamWorking

b. Predictors: (Constant), SkillGapTeamWorking

The following values were calculated for coefficients considering the team working skill gap with productivity improvement. Based on the values calculated in Table 16, the distribution of productivity improvement with the team working skill gap is plotted in the below graph (Figure 11). In addition to that, the R2 line and fit line Y = 1.221 + 0.510 X was also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 16: Coefficients analysis for Team Working

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.221	.100		12.225	.000
	SkillGapTeamWorking	.510	.045	.500	11.332	.000

a. Dependent Variable: ProductivityImprovementTeamWorking

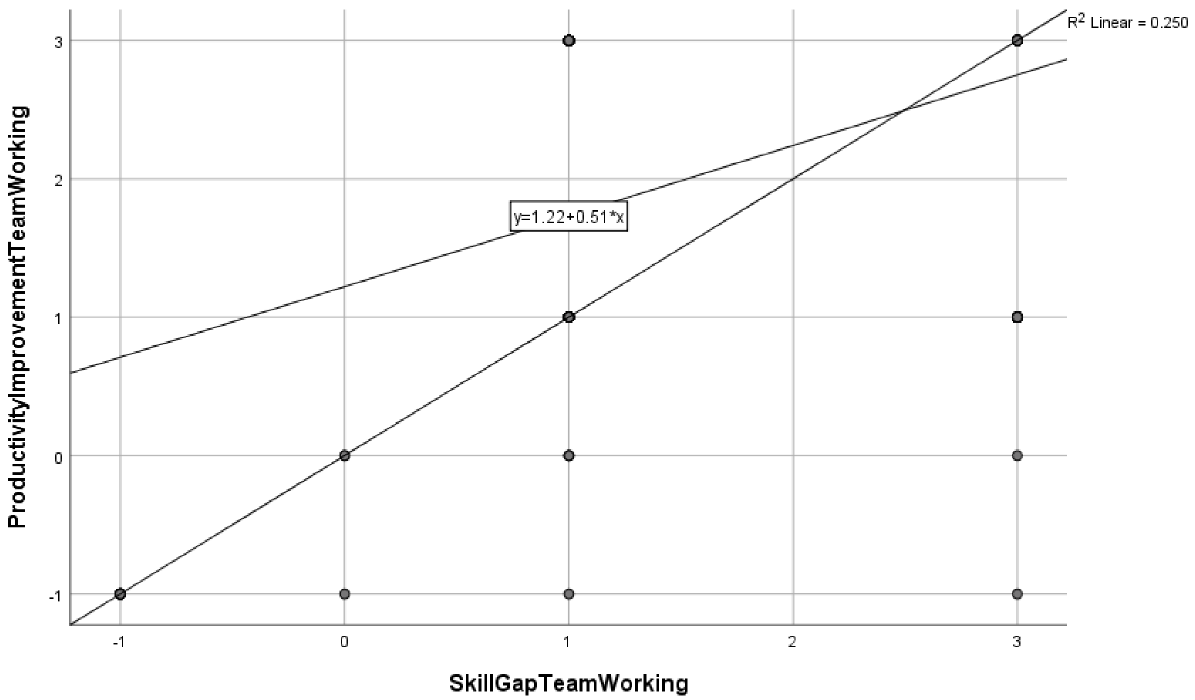

Figure 11: Team Working vs Productivity Graph

Analytical

Table 17 describes model summary details related to productivity improvement (dependent variable) against analytical skill gap(independent variable). Based on the Adjusted R Square value, which is 0.164, 16.4% of the productivity improvement variance can be explained by the analytical skill gap. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 17: Model Summary for Analytical

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.408[a]	.166	.164	1.026

a. Predictors: (Constant), SkillGapAnalytical

Considering ANOVA analysis in Table 18, the analytical skill gap calculated F value is 76.792, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 18: ANOVA analysis for Analytical

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	80.864	1	80.864	76.792	.000[b]
	Residual	405.415	385	1.053		
	Total	486.279	386			

a. Dependent Variable: ProductivityImprovementAnalytical

b. Predictors: (Constant), SkillGapAnalytical

The following values were calculated for coefficients considering the analytical skill gap with productivity improvement. Based on the values calculated in Table 19, the distribution of productivity improvement with the analytical skill gap is plotted in the below graph (Figure 12). In addition to that, the R2 line and fit line Y = 1.380 + 0.445 X were also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 19: Coefficients analysis for Analytical

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.380	.100		13.749	.000
	SkillGapAnalytical	.445	.051	.408	8.763	.000

a. Dependent Variable: ProductivityImprovementAnalytical

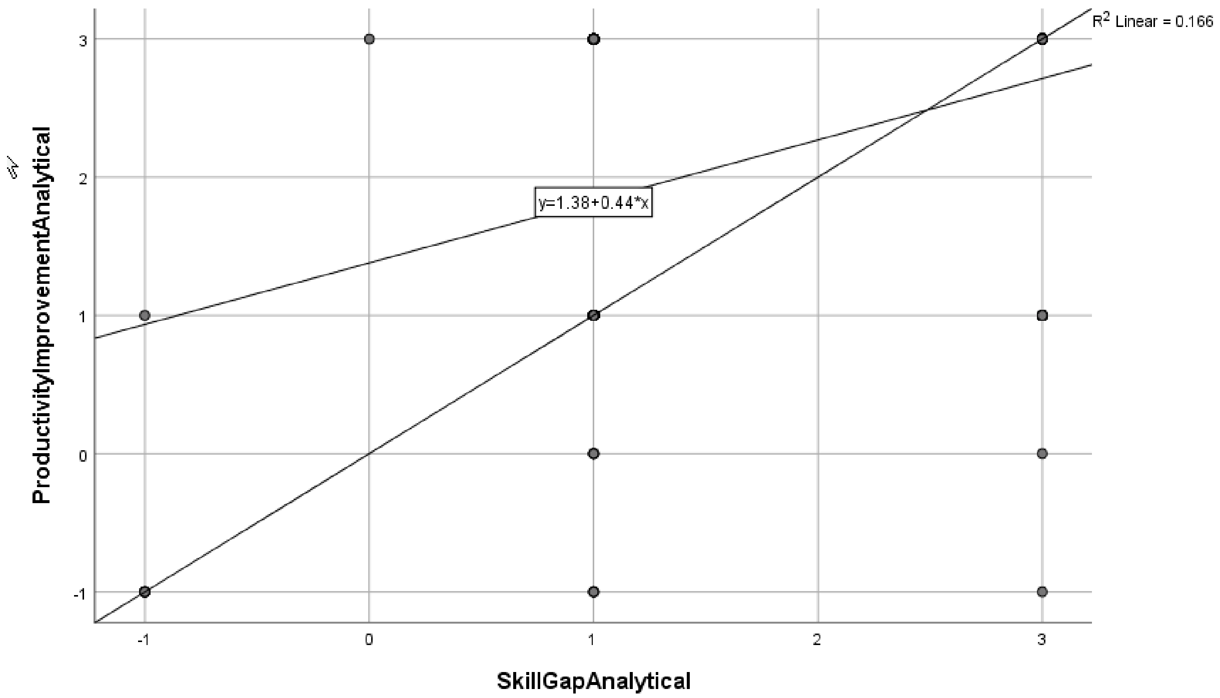

Figure 12: Analytical vs Productivity Graph

<u>Creative Thinking</u>

Table 20 describes model summary details related to productivity improvement (dependent variable) against creative thinking skill gap(independent variable. Based on the Adjusted R Square value, which is 0.153, 15.3% of the productivity improvement variance can be explained by the skill gap of creative thinking. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 20: Model Summary for Creative Thinking

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.394[a]	.155	.153	1.046

a. Predictors: (Constant), SkillGapCreativeThinking

Considering the ANOVA analysis in Table 21, the creative thinking skill gap calculated F value is 70.840, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 21: ANOVA Analysis for Creative Thinking

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	77.521	1	77.521	70.840	.000[b]
	Residual	421.306	385	1.094		
	Total	498.827	386			

a. Dependent Variable: ProductivityImprovementCreativeThinking

b. Predictors: (Constant), SkillGapCreativeThinking

The following values were calculated for coefficients considering the creative thinking skill gap with productivity improvement. Based on the values calculated in Table 22, the distribution of productivity improvement with the creative thinking skill gap is plotted in the below graph (Figure 13). In addition to that, the R2 line and fit line Y = 1.387 + 0.428 X was also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 22: Coefficient Analysis for Creative Thinking

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	1.387	.098		14.216	.000
	SkillGapCreativeThinking	.428	.051	.394	8.417	.000

a. Dependent Variable: ProductivityImprovementCreativeThinking

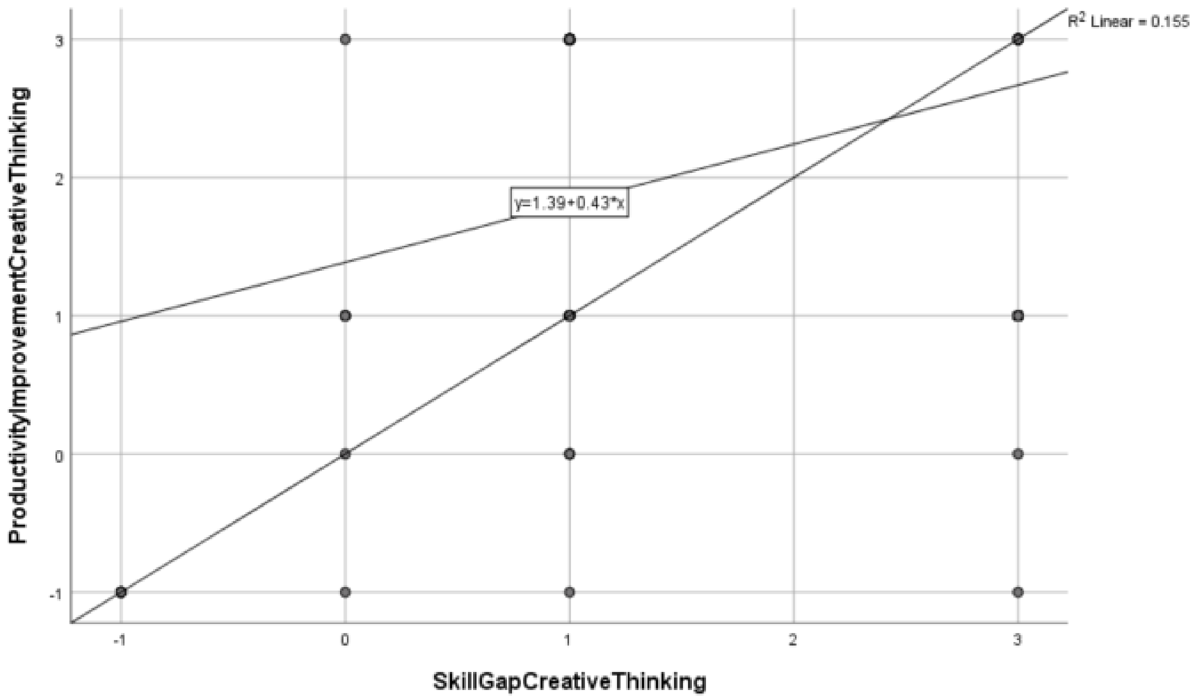

Figure 13: Creative Thinking vs Productivity Graph

<u>Interpersonal</u>

Table 23 describes model summary details related to productivity improvement (dependent variable) against interpersonal skill gap(independent variable). Based on the Adjusted R Square value, which is 0.204, a 20.4% variance in productivity improvement can be explained by the skill gap of interpersonal. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 23: Model Summary for Interpersonal

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.454[a]	.206	.204	1.012

a. Predictors: (Constant), SkillGapInterpersonal

Considering ANOVA analysis in Table 24, the F value for the interpersonal skill gap is calculated as 100.134, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 24: ANOVA Analysis for Interpersonal

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	102.607	1	102.607	100.134	.000[b]
	Residual	394.509	385	1.025		
	Total	497.116	386			

a. Dependent Variable: ProductivityImprovementInterpersonal

b. Predictors: (Constant), SkillGapInterpersonal

The following values were calculated for coefficients considering the interpersonal skill gap with productivity improvement. Based on the values calculated in Table 25, the distribution of productivity improvement with the interpersonal skill gap is plotted in the below graph (Figure 14). In addition to that, the R2 line and fit line Y = 1.113 + 0.523 X was also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 25: Coefficient Analysis for Interpersonal

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	1.113	.097		11.524	.000
	SkillGapInterpersonal	.523	.052	.454	10.007	.000

a. Dependent Variable: ProductivityImprovementInterpersonal

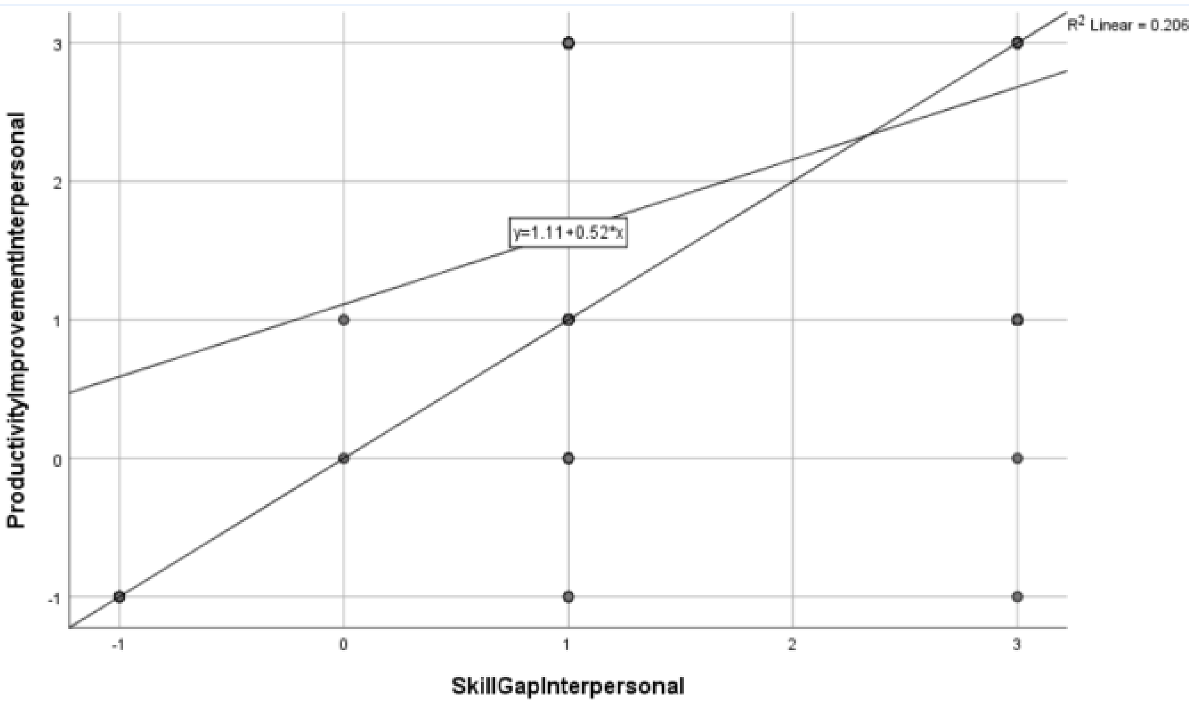

Figure 14: Interpersonal vs Productivity Graph

Time and Task Management

Table 26 describes model summary details related to productivity improvement (dependent variable) against time and task management skill gap(independent variable. Based on the Adjusted R Square value, which is 0.184, 18.4% of the productivity improvement variance can be explained by the skill gap of time and task management. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 26: Model Summary for Time and Task Management

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.431[a]	.186	.184	.993

a. Predictors: (Constant), SkillGapTimeTaskManagement

Considering the ANOVA analysis in Table 27, the time and task management skill gap calculated F value is 88, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 27: ANOVA Analysis for Time and Task Management

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	86.750	1	86.750	88.000	.000[b]
	Residual	379.530	385	.986		
	Total	466.279	386			

a. Dependent Variable: ProductivityImprovementTimeTaskManagement

b. Predictors: (Constant), SkillGapTimeTaskManagement

The following values were calculated for coefficients considering time and task management skill gap with productivity improvement. Based on the values calculated in Table 28, the distribution of productivity improvement with time and task management skill gap is plotted in the below graph (Figure 15). In addition to that, the R2 line and fit line Y = 1.445 + 0.457 X was also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 28: Coefficient Analysis for Time and Task Management

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.445	.095		15.177	.000
	SkillGapTimeTaskManagement	.457	.049	.431	9.381	.000

a. Dependent Variable: ProductivityImprovementTimeTaskManagement

Figure 15: Time and Task Management vs Productivity Graph

Problem Solving

Table 29 describes model summary details related to productivity improvement (dependent variable) against problem-solving skill gap(independent variable). Based on the Adjusted R Square value, which is 0.125, 12.5% of the variance in productivity improvement can be explained by the skill gap of problem-solving. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 29: Model Summary for Problem Solving

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.447[a]	.200	.198	1.003

a. Predictors: (Constant), SkillGapProblemSolving

Considering the ANOVA analysis in Table 30, the problem-solving skill gap calculated F value is 47.126, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 30: ANOVA Analysis for Problem Solving

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	96.793	1	96.793	96.278	.000[b]
	Residual	387.058	385	1.005		
	Total	483.850	386			

a. Dependent Variable: ProductivityImprovementProblemSolving

b. Predictors: (Constant), SkillGapProblemSolving

The following values were calculated for coefficients considering the problem-solving skill gap with productivity improvement. Based on the values calculated in Table 31, the distribution of productivity improvement with the problem-solving skill gap is plotted in the below graph (Figure 16). In addition to that, the R2 line and fit line Y = 1.43 + 0.41 X were also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 31: Coefficient Analysis for Problem Solving

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.366	.098		13.884	.000
	SkillGapProblemSolving	.464	.047	.447	9.812	.000

a. Dependent Variable: ProductivityImprovementProblemSolving

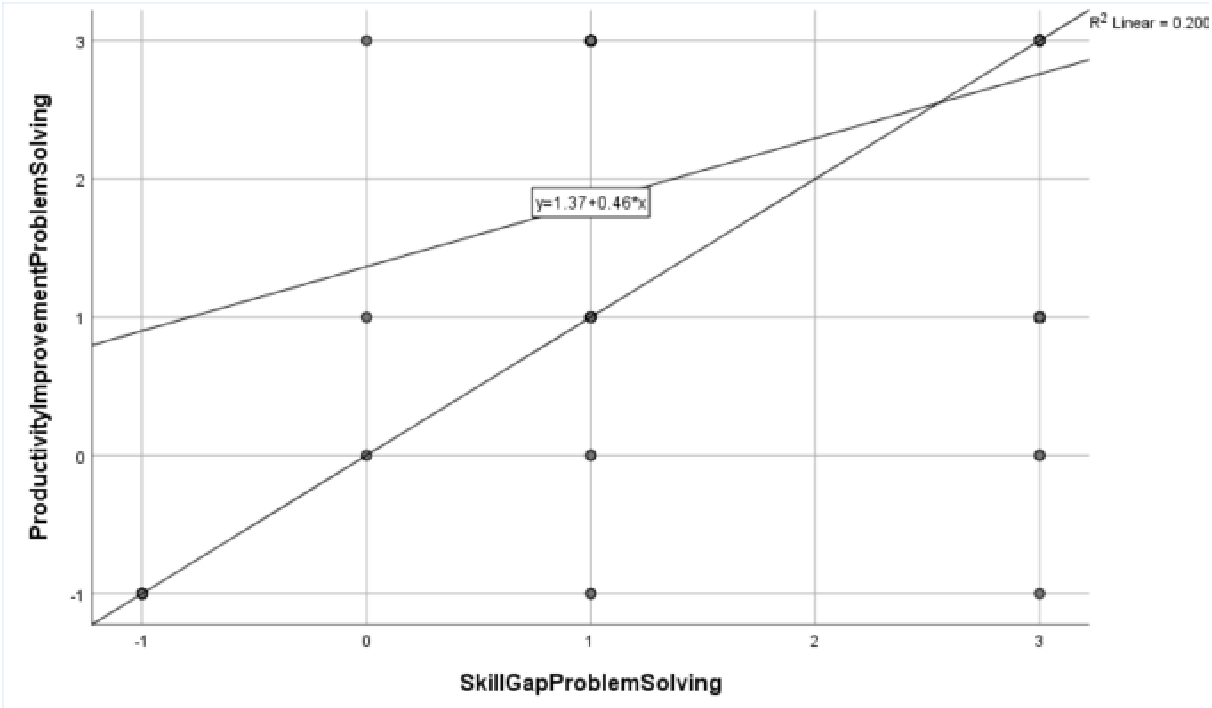

Figure 16: Problem Solving vs Productivity Graph

<u>Programing</u>

Table 32 describes model summary details related to productivity improvement (dependent variable) against programming skill gap(independent variable). Based on the Adjusted R Square value, which is 0.24, 24% of the variance in productivity improvement can be explained by programming's skill gap. However, the relationship level can be considered a medium relationship since the R-value is less than 0.5.

Table 32: Model Summary for Programming

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.492[a]	.242	.240	.978

a. Predictors: (Constant), SkillGapProgramming

Considering ANOVA analysis in Table 33, the programming skill gap calculated F value is 122.813, and the Sig value is .000. Therefore, the null hypothesis has been rejected since the significant value is less than 0.05.

Table 33: ANOVA Analysis for Programming

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	117.396	1	117.396	122.813	.000[b]
	Residual	368.020	385	.956		
	Total	485.416	386			

a. Dependent Variable: ProductivityImprovementProgramming

b. Predictors: (Constant), SkillGapProgramming

The following values were calculated for coefficients considering the programming skill gap with productivity improvement. Based on the values calculated in Table 34, the distribution of productivity improvement with the programming skill gap is plotted in the below graph (Figure 17). In addition to that, the R2 line and fit line Y = 1.284 + 0.511 X were also defined based on Unstandardized Coefficients B values.

(Y = Productivity Improvement and X = Skill gap)

Table 34: Coefficient Analysis for Programming

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	1.284	.094		13.702	.000
	SkillGapProgramming	.511	.046	.492	11.082	.000

a. Dependent Variable: ProductivityImprovementProgramming

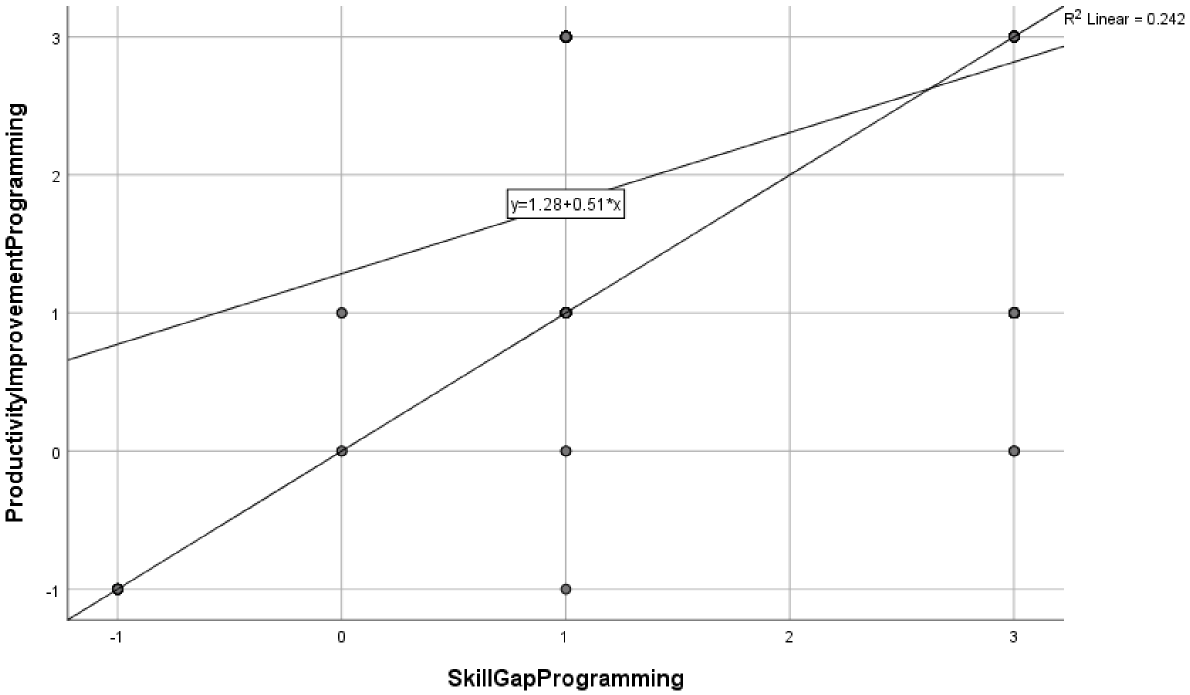

Figure 17: Programming vs Productivity Graph

4.4. Descriptive and regression analysis for samples

After considering the full data set analysis, the next step is to analyze data sets by grouping the data set based on related clusters. Therefore, the following analysis is conducted based on related clusters such as the gender and experience category.

4.4.1. Descriptive and regression analysis for Female Group

As mentioned in the demographic analysis, 55 female responders based on analysis conducted on the group Skill level, Skill gap, and will productivity improve attributes summarized results are mentioned in Table 35.

Table 35: Female group summary

Skill	**Skill Level of the developer (1 to 5 scale)**	**Skill gap with the expectation (Expectation - Skills) (-1 to 3 scale)**	**Will Productivity improve filling the gap? (-1 to 3 scale)**	**Correlation “Skill gap” vs “Will Productivity improves filling the gap?”**
Verbal Communication	4	1.545	2.345	0.257 **Sig(2-tailed) =.058**
Written Communication	4.236	1.673	1.982	0.385 Sig(2-tailed) =.004
Team Working	**4.473**	**2.055**	**2.382**	0.338 Sig(2-tailed) =.012
Analytical	3.982	1.691	2.164	0.358 Sig(2-tailed) =.007
Creative Thinking	3.836	1.455	1.982	0.348 Sig(2-tailed) =.009
Interpersonal	4.109	1.618	2.218	0.450 Sig(2-tailed) =.001
Time and task management skills	4.018	1.691	**2.418**	**0.470** Sig(2-tailed) =.000
Problem Solving	4.091	1.618	2.309	0.374 Sig(2-tailed) =.005
Programming	4.018	1.655	2.2	0.423 Sig(2-tailed) =.001

Considering the above analysis in Table 35, the same as previous analysis on the entire data set, the team working skill type has the highest skill rate, highest skill gap, and the highest productivity improvement score. For verbal communication calculated significant value is 0.058, which is greater than 0.05. Therefore, the null hypothesis cannot be rejected for verbal communication when considering only the female responses. All the skill types have positive correlations when the correlations are calculated for the female group. However, the correlation for all the skills are not strong since those values are less than 0.5

4.4.2. Descriptive and regression analysis for Male Group

As mentioned in the demographic analysis, 332 male responders based on analysis conducted on the group Skill level, Skill gap, and will productivity improve attributes summarized results are mentioned in Table 36.

Table 36: Male Group Summary

Skill	**Skill Level of the developer (1 to 5 scale)**	**Skill gap with the expectation (Expectation - Skills) (-1 to 3 scale)**	**Will Productivity improve filling the gap? (-1 to 3 scale)**	**Correlation "Skill gap" vs "Will Productivity improves filling the gap?"**
Verbal Communication	3.768	1.503	1.988	0.376 Sig(2-tailed) =.000
Written Communication	3.798	1.443	1.816	0.318 Sig(2-tailed) =.000
Team Working	**4.349**	**1.907**	**2.175**	**0.522** Sig(2-tailed) =.000
Analytical	4.051	1.69	2.127	0.415 Sig(2-tailed) =.000
Creative Thinking	3.952	1.636	2.09	0.399 Sig(2-tailed) =.000
Interpersonal	3.952	1.554	1.883	0.455 Sig(2-tailed) =.000
Time and task management skills	3.819	1.651	2.166	0.426 Sig(2-tailed) =.000
Problem Solving	4.123	1.804	2.172	0.461 Sig(2-tailed) =.000
Programming	4.057	1.735	2.16	**0.502** Sig(2-tailed) =.000

Considering the above analysis for all the skill types in Table 36, skill gap and productivity improvement have positive values. Team working skill has the highest skill value, highest skill gap, and highest productivity improvement compared to other skill types.

Based on the correlation calculated in Table 36, All the skill types have a positive correlation. Team working and programing skill gaps strongly correlate with improving productivity since the correlation value is greater than 0.5.

4.4.3. Descriptive and regression analysis for 2-5 Years of IT experience

Considering experience groups in demographic analysis, 2-5 years of experience group has 187 responders. This largest cluster out of other experience-based clusters and an analysis was conducted on Skill level, Skill gap, and will productivity improve attributes summarized results are mentioned in Table 37.

Table 37: 2-5 Years of Experience Group Summary

Skill	**Skill Level of the developer (1 to 5 scale)**	**Skill gap with the expectation (Expectation - Skills) (-1 to 3 scale)**	**Will Productivity improve filling the gap? (-1 to 3 scale)**	**Correlation “Skill gap” vs “Will Productivity improves filling the gap?”**
Verbal Communication	3.802	1.396	1.952	0.366 Sig(2-tailed) = .000
Written Communication	3.829	1.406	1.797	0.4 Sig(2-tailed) = .000
Team Working	**4.358**	**1.856**	2.118	**0.558 Sig(2-tailed) = .000**
Analytical	4.005	1.62	2.07	0.445 Sig(2-tailed) = .000
Creative Thinking	3.941	1.465	1.995	0.387 Sig(2-tailed) = .000
Interpersonal	3.909	1.487	1.904	0.442 Sig(2-tailed) = .000
Time and task management skills	3.872	1.636	2.134	0.435 Sig(2-tailed) = .000

Table 37: 2-5 Years of Experience Group Summary (Continued from previous page)

Problem Solving	4.128	1.711	**2.155**	**0.524** **Sig(2-tailed) = .000**
Programming	4.096	1.706	2.144	**0.538** **Sig(2-tailed) = .000**

Considering the above analysis for all the skill types in Table 37, productivity improvement against all skill gaps has positive values. Teamwork has the highest skill value and the highest skill gap. However, it is not the highest productivity improvement compared to other skill types. In this group, the highest improvement value is calculated for the problem-solving skill type.

According to correlation analysis in Table 37, all the skill types have a positive correlation. There are three strong correlations for teamwork, problem-solving, and programming skill types to improve productivity, which is greater than 0.5.

4.4.4. Descriptive and regression analysis for 0-1 Years of IT experience

Considering the below analysis for all the skill types in Table 38, skill gap and productivity improvement have positive values. Team working skill has the highest skill value and highest skill gap as the other groups. However, it is not the highest productivity improvement compared to other skill types. In this group, the highest productivity improvement value is calculated for time, and task management skill type and team working skill are the second most for improving productivity.

According to the analysis in Table 38, all the skill types have a positive correlation. There are four strong correlations for interpersonal, time and task management, problem-solving, and programming skill types to improve productivity, which is greater than 0.5.

Table 38: 0-1 Years of Experience Group Summary

Skill	**Skill Level of the developer (1 to 5 scale)**	**Skill gap with the expectation (Expectation - Skills) (-1 to 3 scale)**	**Will Productivity improve filling the gap? (-1 to 3 scale)**	**Correlation "Skill gap" vs "Will Productivity improves filling the gap?"**
Verbal Communication	3.594	1.406	1.931	0.369 Sig(2-tailed) =.000
Written Communication	3.851	1.396	1.733	0.347 Sig(2-tailed) =.000
Team Working	**4.297**	**1.861**	2.198	0.479 Sig(2-tailed) =.000
Analytical	3.95	1.604	2.119	0.438 Sig(2-tailed) =.000
Creative Thinking	3.812	1.584	2.099	0.408 Sig(2-tailed) =.000
Interpersonal	3.861	1.564	1.901	**0.566** Sig(2-tailed) =.000
Time and task management skills	3.772	1.614	**2.218**	**0.539** Sig(2-tailed) =.000
Problem Solving	4.02	1.693	2.198	**0.502** Sig(2-tailed) =.000
Programming	3.921	1.634	2.149	**0.576** Sig(2-tailed) =.000

4.5. Descriptive and regression analysis of Other Skills

After analyzing the group data correlation on other variables, the correlation analysis results are mentioned in Table 39 and Table 40.

Table 39: Correlation of Skill Gaps

Correlations

		SkillGapVerbalCommunication	SkillGapWrittenCommunication	SkillGapTeamWorking	SkillGapAnalytical	SkillGapCreativeThinking	SkillGapInterpersonal	SkillGapTimeTaskManagement	SkillGapProblemSolving	SkillGapProgramming
SkillGapVerbalCommunication	Pearson Correlation	1	.612**	.498**	.505**	.409**	.409**	.380**	.452**	.384**
	Sig. (2-tailed)		.000	.000	.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapWrittenCommunication	Pearson Correlation	.612**	1	.438**	.363**	.270**	.405**	.300**	.285**	.295**
	Sig. (2-tailed)	.000		.000	.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapTeamWorking	Pearson Correlation	.498**	.438**	1	.560**	.538**	.581**	.535**	.561**	.523**
	Sig. (2-tailed)	.000	.000		.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapAnalytical	Pearson Correlation	.505**	.363**	.560**	1	.723**	.607**	.599**	.710**	.567**
	Sig. (2-tailed)	.000	.000	.000		.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapCreativeThinking	Pearson Correlation	.409**	.270**	.538**	.723**	1	.597**	.563**	.652**	.566**
	Sig. (2-tailed)	.000	.000	.000	.000		.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapInterpersonal	Pearson Correlation	.409**	.405**	.581**	.607**	.597**	1	.623**	.593**	.566**
	Sig. (2-tailed)	.000	.000	.000	.000	.000		.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapTimeTaskManagement	Pearson Correlation	.380**	.300**	.535**	.599**	.563**	.623**	1	.654**	.579**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000		.000	.000
	N	387	387	387	387	387	387	387	387	387
SkillGapProblemSolving	Pearson Correlation	.452**	.285**	.561**	.710**	.652**	.593**	.654**	1	.737**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000		.000
	N	387	387	387	387	387	387	387	387	387
SkillGapProgramming	Pearson Correlation	.384**	.295**	.523**	.567**	.566**	.566**	.579**	.737**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	
	N	387	387	387	387	387	387	387	387	387

**. Correlation is significant at the 0.01 level (2-tailed).

Considering Table 39, which analyzed the correlation between the skill gap of skill types selected, there is a strong correlation between some skill types and there three skills types as mentioned below which have the highest correlation, which is more than 0.7

- Creative thinking and analytical, 0.723
- Problem-solving and analytical, 0.710
- Programming and problem solving, 0.737

The other interesting fact is that no high correlation (less than 0.5) found with written communication skill except verbal communication skill

Table 40: Correlation of Productivity Improvement

Correlations

		ProductivityImprovementVerbalCommunication	ProductivityImprovementWrittenCommunication	ProductivityImprovementTeamWorking	ProductivityImprovementAnalytical	ProductivityImprovementCreativeThinking	ProductivityImprovementInterpersonal	ProductivityImprovementTimeTaskManagement	ProductivityImprovementProblemSolving	ProductivityImprovementProgramming
ProductivityImprovementVerbalCommunication	Pearson Correlation	1	.758**	.682**	.600**	.588**	.610**	.640**	.597**	.593**
	Sig. (2-tailed)		.000	.000	.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementWrittenCommunication	Pearson Correlation	.758**	1	.641**	.586**	.584**	.646**	.593**	.519**	.552**
	Sig. (2-tailed)	.000		.000	.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementTeamWorking	Pearson Correlation	.682**	.641**	1	.720**	.675**	.683**	.778**	.706**	.668**
	Sig. (2-tailed)	.000	.000		.000	.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementAnalytical	Pearson Correlation	.600**	.586**	.720**	1	.776**	.658**	.686**	.774**	.717**
	Sig. (2-tailed)	.000	.000	.000		.000	.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementCreativeThinking	Pearson Correlation	.588**	.584**	.675**	.776**	1	.643**	.693**	.766**	.726**
	Sig. (2-tailed)	.000	.000	.000	.000		.000	.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementInterpersonal	Pearson Correlation	.610**	.646**	.683**	.658**	.643**	1	.713**	.712**	.652**
	Sig. (2-tailed)	.000	.000	.000	.000	.000		.000	.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementTimeTaskManagement	Pearson Correlation	.640**	.593**	.778**	.686**	.693**	.713**	1	.764**	.725**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000		.000	.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementProblemSolving	Pearson Correlation	.597**	.519**	.706**	.774**	.766**	.712**	.764**	1	.846**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000		.000
	N	387	387	387	387	387	387	387	387	387
ProductivityImprovementProgramming	Pearson Correlation	.593**	.552**	.668**	.717**	.726**	.652**	.725**	.846**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	
	N	387	387	387	387	387	387	387	387	387

**. Correlation is significant at the 0.01 level (2-tailed).

Considering Table 40, which analyzed the correlation between improving software developer's productivity based on selected skill types, every skill type has a strong correlation (greater than 0.5) between each skill type. The highest correlation is calculated for Productivity improvement through problem-solving and programming, which is 0.846.

4.6. Analysis of approaches that provide knowledge and training to improve the productivity

There were three questions added into the survey to respond from the responders related to training/program, which provide knowledge and experience to improve software developers' productivity. Therefore, based on the data collected, a descriptive analysis has been conducted to identify which program/s provide knowledge or experience to improve the software developer's productivity. Calculated data for descriptive analysis is mentioned in Table 41.

Table 41: Descriptive Analysis of Training Stages

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
KnowladeTrainingHigherEducation	387	1	5	3.11	.934
KnowladeTrainingInternship	387	1	5	3.59	1.047
KnowladeTrainingCompany	387	1	5	3.92	1.134
Valid N (listwise)	387				

Based on the above-calculated data in Table 41, the company's training has the highest mean of 3.92. The training provided by the internship comes next with a mean of 3.59, and higher education comes last with a 3.11 mean value.

The next analysis is to determine the correlations of knowledge/training provided on each sector to identify the relationship between them, and calculated details are mentioned in Table 42.

Table 42: Correlation of Training Stages

Correlations

		KnowladeTrainingHigherEducation	KnowladeTrainingInternship	KnowladeTrainingCompany
KnowladeTrainingHigherEducation	Pearson Correlation	1	.400**	.268**
	Sig. (2-tailed)		.000	.000
	N	387	387	387
KnowladeTrainingInternship	Pearson Correlation	.400**	1	.494**
	Sig. (2-tailed)	.000		.000
	N	387	387	387
KnowladeTrainingCompany	Pearson Correlation	.268**	.494**	1
	Sig. (2-tailed)	.000	.000	
	N	387	387	387

**. Correlation is significant at the 0.01 level (2-tailed).

Based on the correlation analysis conducted in Table 42, all the correlations are positive. However, all the relationships are not strong since those correlation values are lower than 0.5. When considering the data's highest correlation, which is 0.494 ~ 0.5, is between the company's training and knowledge and the internship program. The next highest correlation is between the internship program's knowledge and training and higher education, which is 0.4.

4.7. Expert feedback on improving the software developer's productivity

Apart from survey-related questions, interviews were conducted to get expert feedback on improving the software developer's productivity. There were few questions asked from the industrial expert about improving software developer's productivity on the following topics, and the summary of the answers are provided mentioned in Table 43,

1. Which areas should the software developer improve?
2. What training should you provide to improve the productivity of the software developer?
3. When should those trainings provide?
4. What other approaches available?

Table 43: Expert Feedback Summary

Expert Feedback #	**Which areas should improve?**	**What training should be provided?**	**When should the training provide?**	**What other approaches available?**
1	- Understanding the requirement - Problem-solving - Analytic - Code reusability	- Soft skill training	- Before the employment and within the employment	- Self-learning
2	1- Commitment 2- Time Management	1- Improve education to practice commitment 2 - Work on schedules to complete work	1 - During the education 2 - During the employment	- Let the employee decide their carrier path rather than force based on the company requirement

Table 43: Expert Feedback Summary (Continued from previous page)

3	1- Know the core concepts of technologies used 2- Team compassion 3- Communication skills	1- Providing a platform to discuss technical topics 2- Team lead must manage 3 - Providing business communication programs	1 - From starting of the carrier 3 - From education onwards	1- Take part in online courses, and the company can encourage 3- Encourage certifications
4	1- Coding best practices 2- Minimize process ceremonies 3- Understanding the requirement	1- Provide training sessions related to stack 2- Optimizes process ceremonies 3- Training and workshops	1- Continuously during the employment 2- During the employment 3- From education onwards	- Encourage software developers to try out open source projects by the employer
5	1- Time Management 2- Programing skills 3- Team Working skills 4- Communication skills	1,3,4- Training and workshops 2- Innovation sessions	1,3- During the employment 2,4 – From education onwards	- Self-learning by taking online courses

As a summary mentioned in Table 43, there are five expert feedbacks taken from the industrial experts. Further analysis feedback is provided below

4.7.1. Expert feedback 1

Designation – Architect (Java).

Company - Axiata Digital Labs.

Based on his feedback, software developers can improve their productivity by,

- Understanding the requirement for the work that they are doing
- Improving problem-solving skills, analytics skills, and code reusability

Based on his feedback, software developers need to participate in soft skill training to improve the areas mentioned above. Software developers should focus on the course by understanding the requirement of improving soft skills. Also, he suggests that those programs should start before the working environment, but mostly those programs should conduct during the workplace. Further, he mentioned that the software developer should focus on self-learning on technologies.

4.7.2. Expert feedback 2

Designation – Architect (Application).

Company - Virtusa.

Based on his feedback, software developers can improve their productivity by,

- Increasing employee satisfaction
- Commitment to work
- Improving time management

Based on his feedback, work commitment should start from education, and time management should improve from the workplace. He has suggested that the employees should focus on their interest path rather than focusing on the company interest path, and the commitment should improve from the day the software developer start the carrier

4.7.3. Expert feedback 3

Designation – Associate Architect (Java).

Company - SyscoLABS.

Based on his feedback, software developers can improve their productivity by,

- Improving knowledge on the core concept of used technologies
- Good working environment to focus on the job
- Improving business communication skills

Based on his feedback, software developers can improve their productivity by following approaches

- Having a platform share technical knowledge and ideas what they are working which could be useful to others and this should be initiate from the company for an employee to share their knowledge

- Business communication improvement programs should conduct from the education onwards to match the requirement of the industry
- IT companies should promote certifications and online courses by reimbursing and purchasing corporate packages so the software developers can enroll and gain knowledge by doing it. If the company is not promoting those courses, software developers can take those courses to improve their knowledge.

4.7.4. Expert feedback 4

Designation –Architect (.Net, Mobility).

Company - Virtusa.

Based on his feedback, software developers can improve their productivity by,

- Improving coding best practices and provide awareness on those facts such as design patterns, stack-based tools, code quality tools. The employer should provide training continuously those coding best practices related topics
- Minimize process ceremonies and unwanted approval process such as documentation on process-related task and go through the approval process to use development tools
- Improving communication skills to understand the requirement properly. From education onwards, communication skills can be improved, and the employer can provide training as well

Based on his feedback, software developers can improve their productivity by following approaches

- Encourage employees to try out open source projects on the same stack by the employer, so the employee will focus more when the employer is encouraged to do so

4.7.5. Expert feedback 5

Designation –Architect (PEGA).

Company - Virtusa.

Based on his feedback, software developers can improve their productivity by,

- Improving time management, team player skills, and improving leadership skills. To improve those skills, the company can provide training and workshops for the employees during employment and for leadership programs that can specifically target team leads to improve their leadership skills

- Improving communication skills which essential from the beginning of the carrier which should be improved from the higher education and company can provide training and workshops to improve communication skills
- Improving programming skills by conducting innovation sessions to learn new things. Those sessions could start from the higher education, and the company can facilitate the employees to conduct those sessions

Based on his feedback, software developers can improve their productivity by following approaches as well

- Software developers can improve their productivity by self-learning by taking online courses that don't need a classroom to learn, and the software developers can learn new things just sitting in Infront of the computer.

4.8. Summary

In this chapter, the discussion included how the data analysis was conducted on the survey responses. Data analysis was conducted to identify the relationship of the data collected to improve the software developer's productivity. Therefore, a reliability analysis was conducted before analyzing the data on other aspects. Once the reliability test is conducted, Cronbach's alpha was greater than 0.7 to proceed with other data set analyses. Initially, descriptive and regression analysis was conducted to the entire data set to identify the skill gap's relationship to improve the software developer's productivity. Then the same analysis was performed on samples to determine the relationship between data set groups.

5. RECOMMENDATIONS AND CONCLUSION

After analyzing the data collected, the next step is to conclude the data analysis results and provide recommendations on the findings. Therefore, this chapter describes the recommendations and the finding of this research.

5.1. Observations & Recommendations

Based on the analysis conducted in the Data Analysis chapter, there are several observations made. Those observations are described in the following subtopics, and the recommendations are made based on the interview conducted.

5.1.1. Team working skill should improve to improve productivity

Based on the analysis conducted in sections 4.3.1 and 4.3.2.3, the software developers believe that the team working skill is highly rated skill level (4.367 out of 5) and has the highest skill gap (1.928 out of 3). The team working skill has the highest productivity improvement level (2.204 out of 3) as well. When considering the team working skill gap against productivity improvement, it has a strong correlation of 0.5. Therefore, considering those facts, software developers need to improve their team working skills to improve productivity. IT companies can arrange workshops that focus on teamwork and leadership to improve the skill to improve team working skills. Education institutes can add training to their syllabus to improve the software developer's teamwork skills even before the carrier. Based on the expert feedback, these skills can be improved by conducting training by the employer, and the educations institute also can provide that training.

5.1.2. Problem-solving skill should improve to improve productivity

Based on the analysis conducted in section 4.3.1 and 4.3.2.8, Problem-solving skills are considered to be the second-highest skill (4.119 out of 5) rated by the survey responders, and it has a higher skill gap (1.778 out of 3) and productivity improvement (2.191 out of 3) as well. Considering the correlation, it has a relationship, which is not strong. However, it is closed to 0.5, where the correlation is 0.447. Therefore, software developers need to improve problem-solving skills to improve productivity. When it comes to software development problem solving hard to teach in the classroom, it requires specific training and workshops to improve problem-solving skills.

5.1.3. Programming skill should improve to improve productivity

Programming skills are considered an essential skill required for software developers and based on analysis conducted in sections 4.3.1 and 4.3.2.9. It has been calculated to be the third most skill level (4.052/5) of the software developer. Considering the skill gap (1.724), productivity improvement (2.165), and correlation of productivity improvement against skill gap as 0.492, there seems to be a strong relationship (0.492 ~ 0.5) where software developers can improve the programming skills to improve productivity. During software education, they teach software development concepts and technologies. However, software developers must learn new technologies based on the company's requirements for working in the industry. Therefore, based on the expert feedback considering approaches suggested by software developers from the survey, the best way to improve programming skill is to

- Conduct knowledge share sessions within the company and the industry
- Provide a platform among the company or the industry to share knowledge in one platform
- Encourage employees to take certifications from the IT company by reimburse the certification courses and consider them for rating the employees
- Encourage employees to take online courses to improve their programming knowledge. The IT company can purchase corporate packages to encourage employees to take online courses that do day-to-day software development activities.

5.1.4. Written communication skill has less impact on improving software developer’s productivity

Considering skill levels in sections 4.3.1 and 4.3.2.2, written communication in skill level seems to be a high value. However, when considering the written communication skill gap (1.475 out of 3), the productivity improvement (1.84 out of 3) values, and the correlation between skill gap and the productivity improvement (0.33), which is not a strong relationship. Therefore, the written communication has a lower impact on improving software developer's productivity. Since written communication not directly involving improving the productivity of development tasks, the responder may not mark written communication as important for improving productivity. Based on the expert feedback also there was no mention of the written communication.

5.1.5. Verbal Communication, Analytical, Creative Thinking, Interpersonal and Time and task management skills also have an impact on improving software developer’s productivity

As some skills related to this research were mentioned above, other skills involving improving software developer productivity and those skills are mentioned below

- Verbal Communication skills
- Analytical skills
- Creative Thinking skills
- Interpersonal skills
- Time and task management skills

Based on the data analysis conducted in sections 4.3.1 and 4.3.2.1, those skills have averaged around 4 out of 5 skill level ratings. The skill gap is around 1.7 out of 3, and improving productivity also around 2 out of 3. Therefore, even the skill gaps bit lower than the first skills mentioned above, those skills contribute to the software developer's productivity. Based on the expert feedback above, skills are also essential to improve the software developer's productivity. Most industrial experts are pointing out that verbal communication is a vital skill for improving productivity. Even from verbal communication, experts highlight that business communication is essential, and providing training and workshops to improve business communication would be a good step which an IT company can take. Even the software developer is an expert in technical knowledge without good business communication. They may not perform their best when it comes to software development. Apart from that, those training should be conducted before the employment and need to particulate during employment. The best approach would be those programs that could facilitate higher education, and then the IT company should arrange business communication programs for the employees.

5.2. Conclusion

Considering the inputs provided by the responders of the survey and the expert feedback provided on improving software developer's productivity, they believe that there is a potential to improve the software developer's productivity. Based on the survey results, the software developer can improve productivity by improving the following key skills,

- Team Working skills
- Problem-solving skills
- Programming skills

However, other skills considered in the study also contribute to improving the software developer's productivity. However, based on the data analysis conducted in section 4.3.1 and 4.3.2, written communication skill has considered being the skill which has less impact on improving software developer's productivity.

Based on the expert feedback in section 4.7, participating in leadership and teamwork training and communication training will improve the soft skills required for a software developer to improve productivity. Those training should be facilitated during education and employment as well. For communication improvement, industrial experts have recommended providing business communication training covering the communication related to business scenarios during education onwards.

Even though programming skills are considered the key skill for a software developer, to get the most benefit of programming skills and work productively in the industry, the software developer should improve soft skills. The technologies are getting improved in the software development industry, and new technologies are coming to the industry. Therefore, software developers have to learn new things by self-learning, taking online certifications, and trying out new technologies. Based on the expert feedback in section 4.7, this can be facilitated by the IT company and motivate software developers to take online courses as an objective set by the company. Also, software developers can do self-learning on new technologies as well.

Therefore, to improve the software developer's productivity, it is a must to improve programming skills. Only having the programming skills a software developer cannot perform with maximum productivity where software developer has to improve soft skills, including communication, teamwork, and problem-solving skills, get maximum usage of programming skills that a software developer has.

In other words, "Programming skill is the key strength and soft skills such as team working, problem-solving and communication skills support the software developer to perform productively."

5.3. Research Limitations

The sample population was selected from Sri Lanka, and the survey was shared without considering the software developer's designations and the software developer's stream. If the sample population was selected based on the designations or the stream, the responses might change. Suppose the sample was specific to designation levels such as Team Leads. In that case, responses might focus more on soft skills since they have already been in the industry for some time, and their roles and responsibilities may focus on leadership tasks. If the sample was only specific to the .Net stream or java stream, responses might have different technical skills findings.

5.4. Future work

This research's findings can be used for further similar work and recommended to conduct further research on different software development levels such as Team Lead level factors or for the software developers new to the industry. Further, the study can expand to get feedback from the team leads about the junior resources to understand the problem from the leadership perspective.

REFERENCES

Aasheim, C., & Williams, S. (2009). Knowledge and Skill Requirements for Entry-Level Information Technology Workers: Do Employers in the IT Industry View These Differently than Employers in Other Industries?. Journal of Information Systems Education, 5-8, from https://jise.org/Volume20/n3/JISEv20n3p349.pdf

Ahmed, F., Capretz, L.F., Bouktif, S., & Campbell, P.R. (2012). Soft Skills Requirements in Software Development Jobs: A Cross Cultural Empirical Study. J. Syst. Inf. Technol., 14, 58-81, doi:10.1108/13287261211221137

Bailey, J, L. (2014). Non-Technical Skills for Success in a Technical World, ISSN 2219-1933, 2219-6021, doi: 10.30845/ijbss

Berisha-Shaqiri, A. (2015). Impact of Information Technology and Internet in Businesses. Academic Journal of Business, Administration, Law and Social Sciences, 1, 75-78, from http://iipccl.org/wp-content/uploads/2015/03/Ajbals-73-79.pdf

Capretz, L. (2013). Soft Skills and Software Development: A Reflection from Software Industry. International Journal of Information Processing and Management. 4. 171-191, doi:10.4156/ijipm.vol14.issue3.17

Durant, E., Impagliazzo, J., Conry, S., Reese, R., Lam, H., Nelson, V., … McGettrick, A. (2015). CE2016: Updated computer engineering curriculum guidelines. 2015 IEEE Frontiers in Education Conference (FIE). doi:10.1109/fie.2015.7344157

Edwards, J. (2004). 1 Managing Software Engineers and their Knowledge, Springer, 6-7, doi:10.1007/978-3-662-05129-0_1

González-Morales, D., Antonio, L.M., & Garcia, J.L. (2011). Teaching “soft” skills in Software Engineering. 2011 IEEE Global Engineering Education Conference (EDUCON), 630-637, doi:10.1109/EDUCON.2011.5773204.

Graziotin, D., Wang, X., & Abrahamsson, P. (2014). Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ, 23-18, doi:10.7717/peerj.289.

Knobelsdorf, M., & Romeike, R. (2008). Creativity as a pathway to computer science, ACM Sigcse Bulletin. 40. 286-290, doi:10.1145/1597849.1384347

Li, P. L., Ko, A. J., & Zhu, J. (2015). What Makes a Great Software Engineer? 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 20 - 100, doi:10.1109/icse.2015.335

Litecky, C., Aken, A., Ahmad, A., & Nelson, H. (2010). Mining for Computing Jobs. IEEE Software, 27, 78-85, doi:10.1109/MS.2009.150

Matturro, G. (2013). Soft skills in software engineering: A study of its demand by software companies in Uruguay. 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 133-136, doi: 10.1109/CHASE.2013.6614749

Meyer, A.N., Barton, L.E., Murphy, G., Zimmermann, T., & Fritz, T. (2017). The Work Life of Developers: Activities, Switches and Perceived Productivity. IEEE Transactions on Software Engineering, 43, 1178-1193, doi: 10.1109/TSE.2017.2656886

Mtsweni, E., Hörne, T., & Poll, J.A. (2016). Soft Skills for Software Project Team Members. International Journal of Computer Theory and Engineering, 8, 150-155, doi:10.7763/IJCTE.2016.V8.1035

Palacios, R., Caro, E., García-Crespo, Á., & Gómez-Berbís, J.M. (2010). Identifying Technical Competences of IT Professionals: The Case of Software Engineers. Int. J. Hum. Cap. Inf. Technol. Prof., 1, 31-43, doi:10.4018/jhcitp.2010091103

Pavlov, V. L., Busygin, S., Boyko, N., & Babich, A. (2007). Is There Still a Room For Programmers' Productivity Improvement?. In Proceedings of the 5th East-West Design and Test Symposium (EWDTS'07), 146-151

Rajlich, V. (2013). Teaching developer skills in the first software engineering course. 2013 35th International Conference on Software Engineering (ICSE), 1109-1116, doi:10.1109/ICSE.2013.6606661

Sedelmaier, Y., & Landes, D. (2014). Software engineering body of skills (SWEBOS). 2014 IEEE Global Engineering Education Conference (EDUCON), 395-401, doi:10.1109/EDUCON.2014.6826125

Surakka, S. (2007). What subjects and skills are important for software developers? Communications of the ACM, 50, 73 – 78, doi:10.1145/1188913.1188920

Wagner, S., & Ruhe, M. (2018). A systematic review of productivity factors in software development, arXiv preprint arXiv:1801.06475.

APPENDIX A: SURVEY QUESTIONS

A.1. General Question

1.1 How many years have you been working in the IT Industry?

a) 0-1 Years

b) 2-5 Years

c) 6-10 Years

d) More than 10 years

1.2 What is your job role as a programmer?

a) Java development

b) .Net development

d) PHP development

e) C++ development

f) Other

1.3 What is your current position?

a) Senior Tech Lead

b) Tech Lead

c) Associate Tech Lead

d) Senior Software Engineer

e) Software Engineer

f) Associate Software Engineer

g) Other

1.4 What is your gender?

a) Male

b) Female

1.5 What is your age?

a) 20 – 25 years

b) 26 – 30 years

c) 31 – 35 years

d) More than 35

A.2. Skill gap identification questions

1.6 Verbal communication skill level

1.6.1 How would you rate your verbal communication skills?

0, 1, 2, 3, 5

1.6.2 Do your verbal communication skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.6.3 Do you feel that improving verbal communication skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.7 Written communication skill level

1.7.1 How would you rate your written communication skills?

0, 1, 2, 3, 5

1.7.2 Do your written communication skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.7.3 Do you feel that improving written communication skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.8 Team working skill level

1.8.1 How would you rate your team working skills?

0, 1, 2, 3, 5

1.8.2 Do your team working skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.8.3 Do you feel that improving team working skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.9 Analytical skill level

1.9.1 How would you rate your analytical skills?

0, 1, 2, 3, 5

1.9.2 Do your analytical skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.9.3 Do you feel that improving your analytical skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.10 Creative thinking skill level

1.10.1 How would you rate your creative thinking skills?

0, 1, 2, 3, 5

1.10.2 Do your creative thinking skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.10.3 Do you feel that improving your creative thinking skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.11 Interpersonal skill level

1.11.1 How would you rate your interpersonal skills?

0, 1, 2, 3, 5

1.11.2 Do your interpersonal skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.11.3 Do you feel that improving interpersonal skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.12 Time and task management skill level

1.12.1 How would you rate your time and task management skills?

0, 1, 2, 3, 5

1.12.2 Do your time and task management skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.12.3 Do you feel that improving time and task management skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.13 Problem-solving skill level

1.13.1 How would you rate your problem-solving skills?

0, 1, 2, 3, 5

1.13.2 Do your problem-solving skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.13.3 Do you feel that improving problem-solving skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.14 Programming skill level

1.14.1 How would you rate your programming skills?

0, 1, 2, 3, 5

1.14.2 Do your programming skills meet your job role expectations?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.14.3 Do you feel that improving programming skills up to the job role expectation will improve your productivity?

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

A.3. Identify which education/training should improve

1.15 Knowledge and the training provided during higher education are enough to perform productively in the industry.

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.16 Knowledge and the training gained by the internship program are enough to perform productively in the industry.

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

1.17 Training provided by the company is enough to perform productively in the industry.

1) Strongly disagree 2) Disagree 3) Somewhat agree 4) Agree 5) Strongly agree

A.4. Identify Other Skills

1.18 Do you aware of any program or an approach that would improve your productivity?

1) Yes 2) No

1.19 If it is, could you please mention the program/approach

1.20 How will it improve your productivity?

A.5. Question mapping to the variables

#	Variable	Mapping Question number		
1	Verbal Communication skill level	2.1.1	2.1.2	2.1.3
2	Written Communication skill level	2.2.1	2.2.2	2.2.3
3	Team Working skill level	2.3.1	2.3.2	2.3.3
4	Analytical skill level	2.4.1	2.4.2	2.4.3
5	Creative thinking skill level	2.5.1	2.5.2	2.5.3
6	Interpersonal skill level	2.6.1	2.6.2	2.6.3
7	Time and task management skill level	2.7.1	2.7.2	2.7.3
8	Problem-solving skill level	2.8.1	2.8.2	2.8.3
9	Programming skill level	2.9.1	2.9.2	2.9.3
10	Education level	3.1	3.2	-
11	Hands-on experience level	3.1	3.2	3.3

APPENDIX B: INTERVIEW QUESTIONS

A) Do you believe that your employees have the potential to improve their productivity?

1) Yes 2) No

B) If it is ‘Yes’

1.1 What areas would you see that the employees need to improve for maximizing their productivity?

1.2 What kind of training and workshops would you like to suggest which should be provided by the company for the employee to improve those areas mentioned above?

1.3 When would you think those training and workshops should be conducted to maximize employee productivity?

1.4 What are the other approaches that you would like to recommend improving the productivity of the software developer?

C) If it is ‘No’

2.1 Why do you think that your employees do not have the potential to improve their productivity?