IT STAFFS' TENDENCIES TOWARD CHOOSING EFFORT SIZING MEASURE:

MAN/DAY vs STORY POINTS

ŞEVKET TEBER

BOĞAZİÇİ UNIVERSITY

2023

IT STAFFS' TENDENCIES TOWARD CHOOSING EFFORT SIZING MEASURE:
MAN/DAY vs STORY POINTS

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Şevket Teber

Boğaziçi University

2023

IT Staffs' Tendencies Towards Choosing Effort Sizing Measure:

Man/Day vs Story Points

The thesis of Şevket Teber

has been approved by:

Prof. Meltem Özturan
(Thesis Advisor)

Assist. Prof. Nazım Taşkın

Prof. Birgül Kutlu Bayraktar
(External Member)

June 2023

DECLARATION OF ORIGINALITY

I, Şevket Teber, certify that

• I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;

• this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;

• this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature..

Date ....

ABSTRACT

IT Staffs’ Tendencies Towards Choosing Effort Sizing Measure:

Man/Day vs Story Points

Effort estimation is crucial for the successful management of software development projects. The man-day and story points techniques are widely employed by professionals to gauge the necessary resources for a project's completion. This study aims to explore the differences between the man-day and story points techniques in the contexts of experience, flexibility, complexity, teamwork, and performance tracking. The potential distinctions that made impact the effectiveness of each approach is investigated utilizing a survey of professionals who use either of these two techniques.

Contrary to expectations, the analysis does not reveal significant differences between the man-day and story points techniques in terms of their applicability to various project dimensions like need for expertise, flexibility, complexity, teamwork and performance tracking. Both techniques show similar results in the contexts of experience, flexibility, complexity, teamwork, and performance tracking. These findings suggest that the choice of estimation technique may be more dependent on individual preferences, team culture, or specific project requirements rather than inherent advantages or disadvantages of the techniques themselves.

Given the lack of substantial differences between the techniques, practitioners are encouraged to select an estimation method that best aligns with their team's experience and working style, as well as the nature of the software development project.

ÖZET

BT Çalışanlarının Eforlama Ölçütü Seçmeye Yönelik Eğilimleri:

Adam/Gün - Story points Karşılaştırması

Efor tahmini, yazılım geliştirme projelerinin başarılı yönetimi için çok önemlidir. Adam-gün ve story points teknikleri, bir projenin tamamlanması için gerekli kaynakları ölçmek için profesyoneller tarafından yaygın olarak kullanılır. Bu çalışma, deneyim, esneklik, karmaşıklık, ekip çalışması ve performans izleme bağlamlarında adam-gün ve story points teknikleri arasındaki farkları keşfetmeyi amaçlamaktadır. Bu iki teknikten birini kullanan profesyonellerle yapılan bir anketten yararlanarak, her bir yaklaşımın etkinliğini etkileyebilecek potansiyel farklılıkları araştırılmıştır.

Beklenenin aksine proje uygulanabilirliği açısından adam-gün ve story points teknikleri birbiri arasında önemli farklılıklar ortaya koymamaktadır. Her iki teknik de deneyim, esneklik, karmaşıklık, ekip çalışması ve performans takibi bağlamında benzer sonuçlar göstermektedir. Bu bulgular, tahminleme tekniği seçiminin, tekniklerin kendilerine özgü avantajları veya dezavantajlarından ziyade bireysel tercihlere, ekip kültürüne veya özel proje gereksinimlerine daha fazla bağlı olabileceğini ortaya koymaktadır.

Teknikler arasında önemli farklar olmadığı göz önüne alındığında, uygulayıcılar ve proje yöneticileri, ekiplerinin deneyimi ve çalışma tarzının yanı sıra yazılım geliştirme projesinin doğasına en uygun tahmin ölçüm birimini seçmeye teşvik edilmektedir.

ACKNOWLEDGMENTS

I am delighted to recognize the essential individuals who played a significant role in the successful completion of my dissertation.

I would like to thank my thesis advisor Prof. Dr. Meltem Özturan for unwavering support, guidance, and mentorship throughout this study. Her expertise, encouragement, and valuable insights have been instrumental in shaping my research.

I would also like to extend my appreciation to Assist. Prof. Nazım Taşkın and Prof. Birgül Kutlu Bayraktar for their invaluable contributions as committee members.

Lastly, I wish to express my sincerest thanks to my beloved fiancée, soon to be my wife, Aybelen Çelebi. Her constant love, support and understanding have been my source of strength and inspiration during this challenging journey.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: THEORETICAL BACKGROUND.. 3

2.1 Software development life cycle and estimation techniques 3

2.2 Overview of story points and man-day sizing measures................................. 7

2.3 Comparison and evaluation of story points and man-day sizing measures 7

CHAPTER 3: LITERATURE REVIEW 10

3.1 Experience and expertise in effort estimation 10

3.2 Flexibility and adaptability in effort estimation.. 12

3.3 Collaboration and teamwork in effort estimation 13

3.4 Complexity and uncertainty in effort estimation 15

3.5 Performance tracking and management in effort estimation 17

CHAPTER 4: RESEARCH METHODOLOGY 20

4.1 Research design.. 20

4.2 Participants and sampling... 21

4.3 Data collection……………………………………………………………… 22

4.4 Hypotheses……………………………………………………….................22

4.5 Data analysis...23

CHAPTER 5: RESULTS AND FINDINGS.. 25

5.1 Demographic characteristics of the sample... 25

5.2 Factor analysis.. 32

5.3 Reliability analysis 39

5.4 Independent Samples T-Test... 41

CHAPTER 6: CONCLUSION.. 46

6.1 Discussion of the results 46

6.2 Implications of the results..... 50

6.3 Limitations and future research suggestions 50

APPENDIX: SURVEY QUESTIONS... …52

REFERENCES 61

LIST OF TABLES

Table 1. Age Distribution of the Sample ... 27

Table 2. Experience Distribution of the Sample ... 28

Table 3. Positions Distribution of the Sample ... 29

Table 4. Education Level Distribution of the Sample ... 30

Table 5. Gender Distribution the Sample ... 31

Table 6. Sizing Measure Distribution the Sample ... 31

Table 7. Pattern Matrix of Cross Loading Items ... 33

Table 8. Pattern Matrix of Survey Items - 1 ... 34

Table 9. Pattern Matrix of Survey Items - 2 ... 35

Table 10. Pattern Matrix of Survey Items - 3 ... 36

Table 11. Pattern Matrix of Survey Items - 4 ... 37

Table 12. Pattern Matrix of Survey Items - 5 ... 38

Table 13. Pattern Matrix of Survey Items - 6 ... 38

Table 14. Cronbach's Alpha Values of Items of Determined Subfactors for each main factor ... 40

Table 15. Independent Samples T-tests of Main Factors ... 42

Table 16. Independent Samples T-tests of Determined Subfactors ... 44

CHAPTER 1

INTRODUCTION

Software systems play a pivotal role in many aspects of the modern world. In order to create software, a method known as the software development life cycle (SDLC) is often used. Phases of planning, analysis, design, implementation, testing and maintenance are frequently included in the process steps (Leau, Loo, & Tham, 2012). Each phase is critical to ensuring that the program is created appropriately and satisfies the client's expectations. The planning phase of the software development life cycle is the first stage. The project team will establish the project's scope at this phase, as well as develop a schedule and budget. This phase is crucial for making sure the project is finished on schedule and within budget (Lester, 2006). Estimating how much work has to be done is one of the essential activities that cannot be passed over in the planning stage of a project. This process is known as effort estimation.

To elaborate, effort estimation is the procedure of calculating an estimate of the time and resource that are required to finish a given project (Turner, 1999). Methods like the Story points, Man-Day, Delphi Method, Function Point Analysis, Use Case Points and Wideband Delphi are frequently employed for this purpose. These methods allow project managers and team members to better plan and manage their endeavors by having a more precise idea of how much time and effort are needed for seeing them through to completion.

The development teams' behavior and the results of their work are both affected by the estimation methods that they utilize during the production process. There is a possibility of observable behavioral variations occurring between IT employees who utilize different metrics of size. For instance, many software development teams prefer to use estimation techniques that rely on experts' subjective assessment, in other words, their experience (Hughes, 1996). In addition, while some techniques apply the exact time to spend on a task or a project, some techniques use mostly series applied sizing measure which makes it more complex. Also, team members have to agree about the size of the task, so it requires solid teamwork (Schwaber & Beedle, 2001). Due to the nature of some techniques, the performance of the team member cannot be traced with how much sizing measure metrics the member works, thus it may affect performance tracking. In some techniques change in estimations does not immediately can be associated to the time that is needed, hence it affects flexibility of team or team members.

Despite the widespread use of man-days and story points method in software development, there is a lack of research comparing the two techniques. This research aims to explore behavioral tendencies of IT staff when choosing either story points or man-day effort sizing measurement. The research will especially focus on the comparison of story points and man-day techniques in terms of their ability of importance of experience, the flexibility, complexity, need of teamwork and performance tracking, as well as their corresponding sub factors that affects them.

RQ1: How do the tendencies of professionals differ in terms of experience, flexibility, teamwork, complexity, and performance tracking when using the man-day approach compared to the story points approach?

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Software development life cycle and estimation techniques

The Software Development Life Cycle (SDLC) is a systematic process for developing software that ensures its quality (Alshamrani & Bahattab, 2015). The SDLC consists of several well-defined phases, each with specific tasks and objectives:

i. Planning: During this phase, clients provide specifics and concepts for their software, and the organization plans the required resources, time, and budget for the proposed development (Lekh, 2015).

ii. Analysis: In this phase, stakeholders' needs and requirements are gathered, analyzed, and documented in a comprehensive and unambiguous manner (Dora & Dubey, 2013).

iii. Design: During this phase, the software requirements are turned into a classification of software modules and their boundaries (Sharma & Pandey, 2015).

iv. Development: This phase involves the actual coding/programming and software program integration (Lekh, 2015).

v. Testing: The software is thoroughly tested to ensure that it is free from defects and meets the desired quality standards (Jindal, 2016).

vi. Maintenance: After-delivery software change to fix bugs, improve performance, or adapt to new conditions (Jansson, 2007).

Project planning is a crucial aspect of managing projects, involving the selection of enterprise objectives and the establishment of necessary policies, procedures, and programs to achieve them. Effective planning in a project environment entail setting a predetermined course of action within a forecasted context, with upper-level management actively involved in selecting alternatives during the planning stage. Successful project planning requires a systematic, flexible, disciplined approach that is capable of accepting multifunctional inputs, as it is an iterative process performed throughout the project's life. The primary objectives of project planning are to eliminate or reduce uncertainty, improve operational efficiency, obtain a better understanding of objectives, and provide a basis for monitoring and controlling work. Inadequate planning can lead to negative consequences, such as poorly defined requirements and a chaotic project environment. Systematic planning helps organizations set goals and avoid reactive management, which can result in crisis management and conflict (Kerzner, 2000).

Estimation techniques play a vital role in the project planning phase, as they directly influence the allocation of resources, time management, and overall project success (Turner, 2009). These techniques serve as the foundation for determining the effort required to complete tasks, allowing for better organization and prioritization of project activities. By accurately estimating the effort required for various tasks, project managers can create realistic schedules and allocate resources efficiently, ultimately leading to improved productivity and streamlined project execution.

Incorporating estimation techniques into the planning phase enables organizations to take a proactive approach in mitigating risks and addressing potential challenges. These techniques provide a quantitative basis for decision-making and foster collaboration and transparency among team members. By integrating experience, flexibility, complexity, teamwork, and performance tracking factors into these estimation methods, project managers can enhance the overall effectiveness of their project planning. A comprehensive understanding of estimation techniques helps organizations achieve their objectives more efficiently, reducing the likelihood of setbacks and facilitating adaptive responses to changing circumstances. Ultimately, the effective use of estimation techniques contributes to better project outcomes and enables organizations to deliver high-quality results in a timely manner.

Agile development methodologies have gained prominence as an alternative to traditional software development life cycle methodologies due to their iterative, incremental development processes that prioritize collaboration, flexibility, and customer satisfaction (Al-Saqqa, Sawalla, & AbdelNabi, 2020). Examples of agile methodologies include Scrum, Extreme Programming (XP), and Kanban, which emphasize continuous delivery, adaptive planning, and the ability to embrace change (Saleh, Huq, & Rahman, 2019).

Scrum focuses on iterative and incremental progress through time-boxed sprints. It emphasizes collaboration, communication, and flexibility, with roles such as Product Owner, Scrum Master, and Development Team. Scrum relies on regular meetings, or ceremonies, including sprint planning, daily stand-up, sprint review, and sprint retrospective, to ensure progress and adapt to changes (Schwaber & Beedle, 2001). Extreme Programming emphasizes continuous improvement, flexibility, simplicity, and high-quality software delivery. XP practices include continuous integration, test-driven development, pair programming, and small, frequent releases. These practices aim to improve software quality, enhance collaboration, and adapt to changing requirements (Beck, 2000). Kanban is an agile method for managing work in progress by visualizing the workflow, limiting work in progress (WIP), and continuously improving the process. It originated from the Toyota Production

System and has been adapted for software development and other knowledge work. Kanban focuses on transparency, flexibility, and collaboration, using a visual board to represent tasks as they move through various stages of completion (Anderson, 2010).

In contrast to traditional SDLC methodologies such as Waterfall, agile approaches focus less on upfront planning and documentation and more on rapid delivery of working software (Dyba & Dingsoyr, 2008). This enables agile teams to respond more effectively to changing requirements and market conditions, resulting in improved project outcomes and reduced development risk.

A crucial aspect of agile development is the use of estimation techniques to predict the effort required to complete software development tasks. Estimation in agile methodologies supports effective planning, tracking, and control of the development process. Several estimation techniques have been proposed for use in agile development, which can be briefly overviewed as follows:

Story points: This method involves estimating the relative effort required to complete a user story or task, considering factors such as complexity, uncertainty, and the amount of work needed (Cohn, 2005).

Man-Day: Man-day estimation represents the amount of work that can be completed by one person in one day, providing a more explicit measure of effort in terms of time (Ferguson, 1971).

T-Shirt Sizing: This method uses t-shirt sizes (e.g., Small, Medium, Large, and Extra-Large) to represent the relative effort required to complete tasks, providing a simple and intuitive way to estimate work (Mallidi & Sharma, 2021).

2.2 Overview of story points and man-day sizing measures

The processes of planning, scheduling and delivering a project need an accurate estimation of the amount of effort. Accurate effort estimation can assist teams in planning their work, establishing goals that are attainable, and delivering software projects on time and under budget. However, effort estimation is not always an easy task. Estimates can be made more accurate depending on a variety of factors including expertise, teamwork, adaptability and complexity.

Effort estimation in software development can often be done using the man-day sizing method or the story points sizing method. The man-day approach calculates the amount of labor necessary to finish a work based on how many days it will take to finish (Ferguson, 1971). On the other hand, the amount of a story's point is determined on how much work, effort, and risk it might take during the process of development (Mallidi & Sharma, 2021).

Each method has benefits and drawbacks. The suitability of each for a particular project depends on factors including the level of flexibility, the nature of the task at hand, and the level of team collaboration (Fehlman & Santillo, 2010). Project managers, developers, and all other stakeholders in software development projects must have a thorough awareness of the differences between these techniques.

2.3 Comparison and evaluation of story points and man-day sizing measures

Effort estimation in software development has been a subject of research for several decades. In recent years, the focus has shifted towards agile development methodologies and their associated estimation techniques. Two widely used sizing measures in the agile development process are story points and man-days.

Story points is a unit of measure for the complexity and effort required to complete a user story or task in a project. They are a relative measure, as they do not correspond directly to time or effort, but rather provide a means of comparison between different tasks (Cohn, 2005). Man-days, on the other hand, represent the number of working days required to complete a task, with one man-day equating to a day's worth of work for one person (Ferguson, 1971).

In terms of strengths, story points provide a more abstract and flexible measure of effort estimation, allowing teams to account for the varying levels of complexity and uncertainty inherent in software development projects (Cohn, 2005). This flexibility can be particularly beneficial when working with multidisciplinary teams, where differences in skill sets, experience, and domain knowledge can impact the effort required to complete a task (Deemer, Benefield, Larman, & Vodde 2012).

On the other hand, man-days provide a more concrete measure of effort estimation, which can be beneficial for tracking and managing the progress of a project (Zivadinovic, Medic, Maksimovic, Damnjanovic, & Vujcic, 2011). The man-day estimation can be used for comparisons between estimated and actual effort, and later it can be used for variance analysis. This data generated by the variance analysis are useful for management control and decision making in areas such as resource allocation, outsourcing of specific development activities, and learning from adoption of new development tools and practices (Ooi & Soh, 2003). It can be essential element in the calculation of estimated project costs and schedule.

When comparing these two measures, it is important to consider the context in which they are being used. For example, in an agile environment where flexibility and adaptability are key, story points may be more suitable, as they allow for adjustments to be made as the project progresses (Deemer et al., 2012). Conversely, in a more traditional software development process where strict timelines and schedules are paramount, man-days can be the preferred measure.

Ultimately, the choice between story points and man-days will depend on the specific project and team dynamics, as well as the goals and priorities of the organization. Research has shown that the most accurate and reliable estimates are achieved by leveraging the strengths of more than one sizing measures and combining them in a hybrid approach (Fischer, 1981).

CHAPTER 3

LITERATURE REVIEW

Software development estimation methods are richly researched domain, and this literature review intends to delve into it, focusing on the comparative exploration of man-day and story points estimation techniques. The review is organized into five sections, each detailed into a distinct but connected aspects that related to the process of software development effort estimation.

The study initially explores the impact of experience in effort estimation, outlining their influence on the accuracy of time predictions for task completion. In second section, the importance of flexibility and adaptability due to the ever-changing landscape of software projects is highlighted. The role of teamwork and collaboration is underscored, detailing their contribution to more precise and trustworthy estimates. The literature review also addresses the obstacles posed by complexity and uncertainty in software development projects. Finally, it concluded by emphasizing the crucial role of performance tracking in relation to effort estimation, ultimately providing a broad framework for interpreting and understanding the study's findings.

3.1 Experience and expertise in effort estimation

When software development tasks are precisely estimated, it is easier to make better plans for timelines, resources, and finances (Boehm, 1984). The accuracy of the effort estimates used in software projects is critical to the timely completion of those projects. The team members' experience is crucial in predicting the time needed to complete a task.

Many studies have investigated how experienced experts might be able to assist with the estimation of software projects. It was demonstrated by Jørgensen that the use of expert opinion is more accurate in the estimation of effort than the use of other estimation models (2004). The study indicated that using expert opinion in conjunction with other methods of estimating yielded the greatest amount of valuable information. Niessink and Van Vliet (1997) used an industrial software system as an example, during which they observed 140 change tasks and compared the estimates provided by the original experts to the estimates provided by formal estimating models (function points and analogy). The estimates that were generated by professionals had a higher degree of precision compared to those that were made utilizing function points.

According to Tanveer, Guzman, & Engel (2017), having relevant implementation experience contributes positively to estimation accuracy. In contrast, they found that estimates made for similar backlog items in previously completed sprints were not as important or relevant compared to other factors. This finding suggests that the number of estimated projects might have a different impact on the estimation accuracy than implementation experience.

Furthermore, Jorgensen & Grimstad (2012) indicated that both experienced software professionals and those with varying self-assessed skill levels are susceptible to estimation biases concerning interdependent tasks. Even though experienced professionals understand complexities better, they can still be biased. Highly skilled professionals may underestimate the resources needed due to confidence in their abilities, while those with lower self-assessed skills might overestimate the requirements for projects with low interdependence.

As a result, having the necessary expertise and experience is essential for estimating the amount of work required for software development. Only experienced software engineers and teams can accurately estimate the work needed to complete a project successfully.

3.2 Flexibility and adaptability in effort estimation

The nature of software projects, which is constantly evolving, makes it hard to accurately predict the amount of effort required. A project's needs, technology, and overall scope may change significantly over time, impacting the time and resources required to execute it. In order to get a reliable estimation, flexibility has to be considered while planning the project.

Development flexibility is the ability to make changes to a product without incurring significant additional costs. Thomke & Reinertsen (2012) propose that development flexibility is a better approach than precise long-term forecasting because it allows for more adaptability. Managers can use design technologies and management methods to increase the "turning radius" of their development team, reducing their reliance on exact long-term forecasting and lowering their risk exposure. To reduce the number of necessary changes, one approach is to decrease the development cycle time. However, it's equally important to minimize the costs associated with necessary adjustments. This work explores how to limit the damage caused by inaccurate forecasts and argues that adaptation can be crucial in achieving this goal.

According to research by Molokken & Jorgensen (2005), projects using a flexible development model (such as evolutionary or incremental) were more likely to experience effort overruns of a smaller scale than comparable projects using a sequential (waterfall) development model. Adaptability to changes and robustness are also related to flexibility. Research has shown that agile methods promote adaptability to changes, allowing teams to respond effectively to evolving requirements (Begel & Nagappan, 2007). Robustness refers to the ability of a software development process to withstand variations in its environment and is considered an essential characteristic for effective software development, Jayaswal & Patton (2006) suggest that robust software development model should be implemented in order to ensure trustworthy software.

The diverse findings from studies suggest that it is essential to include flexibility in the process in order to increase the accuracy of estimation in development projects. By considering the possibility of modifications to the project's needs and their capacity to handle those changes, development teams can improve their accuracy. Teams can complete tasks more quickly and be better prepared for any necessary revisions with this strategy.

3.3 Collaboration and teamwork in effort estimation

Effective teamwork and collaboration play a critical role in the success of software development projects, particularly in the context of effort estimation (Deemer et al., 2012). Software effort estimation is generally conducted as a cooperative process where software experts from various disciplines merge their unique skills and knowledge to complete a specific task (Børte & Nerland, 2010). Agile methodologies, such as Scrum and Extreme Programming, emphasize the importance of communication, collaboration, and shared decision-making among team members to achieve more accurate and reliable effort estimates (Schwaber & Beedle, 2001).

In 2004, Moløkken-Østvold and Jørgensen conducted a study on the effectiveness of group discussions in comparing and discussing estimates that were previously made on an individual basis. The study found that estimates became more accurate after group discussions compared to the average of the individual estimates. This indicates that estimates made with teamwork have more accuracy than individual estimates.

Jørgensen (2004) conducted a large industrial study that compared two estimation approaches, namely top-down and bottom-up, to estimate software projects. The study found that the bottom-up approach, which involves breaking down the project into different activities and estimating their efforts separately before adding them up, generally resulted in more accurate estimates. Additionally, the study found that almost half of the time (49%) was spent on discussing the requirements and context of the project, as well as breaking down the project into activities.

Drescher & Garbers (2016) found that higher commonality within a team can contribute to higher intended performance by fostering enhanced communication, understanding, trust, and collaboration among team members. This improved environment allows for more effective decision-making and increased motivation, which ultimately leads to better performance. Lingard & Berry (2002) identify participation as one of the critical factors affecting group performance in software engineering projects. Active participation from all team members is essential for efficient communication, decision-making, and problem-solving, which are crucial for project success. Tanveer et al. (2017) considered estimates made for a similar BLI in previously completed sprints, which can be interpreted as lessons learned, as an important factor to improve estimation accuracy.

In conclusion, teamwork and collaboration are essential components of agile development methodologies and contribute significantly to the accuracy of effort estimation. By incorporating collaborative estimation techniques, teams can leverage diverse perspectives and expertise, reducing the impact of cognitive biases and producing more accurate and reliable estimates.

3.4 Complexity and uncertainty in effort estimation

Complexity and uncertainty are inherent aspects of software development projects, posing significant challenges to the process of effort estimation (Boehm, 2012). As the scope and complexity of tasks increase, accurately estimating the required effort becomes increasingly difficult. To cope with this, tasks should be divided into smaller pieces. (Jorgensen, 2004). Similarly, uncertain factors, such as changing requirements, technological constraints, and personnel changes, contribute to the challenge of producing reliable estimates (Glorath & Evans, 2006).

Agile development methodologies, with their focus on iterative and incremental development, are designed to better handle complexity and uncertainty (Highsmith & Cockburn, 2001). By breaking projects into smaller, manageable tasks and continuously reassessing and adjusting effort estimates, development teams can more effectively respond to changes and uncertainties that arise during the software development process (Beck, 2000).

Several techniques have been proposed to address the challenges posed by complexity and uncertainty in effort estimation. One such approach is the use of expert judgment, which leverages the knowledge and experience of domain experts to produce more accurate estimates (Jørgensen & Sjøberg, 2004). Another approach involves the use of historical data to compare the current project with similar past projects to derive effort estimates (Kumar, Behera, Kumari, Nayak, & Naik 202).

Research has shown that incorporating uncertainty into effort estimation can lead to more reliable results. For example, Jørgensen and Grimstad (2011) found that providing developers with information about the uncertainty of their estimates led to better calibration and improved accuracy. In addition, studies have suggested that using a combination of estimation techniques can produce more accurate and reliable estimates, as different methods may be better suited to handle different aspects of complexity and uncertainty (Moløkken-Østvold et al., 2008).

Zahraoui & Idrissi (2015) suggested that when estimating with story points, the complexity factor should be considered. This complexity factor includes dependencies on other systems and the clarity of the task, which can be interpreted as the uncertainty of the task. Furthermore, Tanveer et al. (2017) emphasized that underestimating the complexity of backlog items is a common challenge in agile software development projects. This underestimation can lead to inaccurate effort estimations, resulting in project delays, cost overruns, and reduced overall project success.

In conclusion, complexity and uncertainty pose significant challenges to the process of effort estimation in software development projects. Agile development methodologies, along with a combination of estimation techniques, can help address these challenges, allowing teams to more effectively manage and respond to the complex and uncertain nature of software projects.

3.5 Performance tracking and management in effort estimation

Performance tracking and management are crucial components of agile development methodologies, as they enable teams to monitor progress, evaluate the effectiveness of their estimates, and adjust their plans accordingly (Cohn, 2006). By regularly monitoring progress and comparing actual effort to initial estimates, agile teams can identify areas where their estimation practices may need refinement and take corrective actions to improve future estimates (Rawsthorne, 2008).

One approach to performance tracking and management is the use of burndown charts, which display the remaining work for a project or iteration in relation to the time remaining (Cohn, 2006). Burndown charts serve as a valuable tool for teams to visualize their progress and identify deviations from initial estimates, enabling them to make necessary adjustments and maintain alignment with project goals. By providing a clear representation of the remaining work in relation to the time available, burndown charts help teams stay on track and make informed decisions about the allocation of resources and effort (Chaudhuri & Chaudhuri, 2011).

Velocity, another performance metric commonly used in agile development, measures the rate at which a team completes work, typically expressed as story points per iteration (Cohn, 2005). By tracking and comparing their velocity over multiple iterations, teams can gain insights into the consistency and accuracy of their effort estimates, allowing them to refine their estimation processes and improve overall project predictability (Rubin, 2012).

Research has shown that performance tracking and management can have a significant impact on the accuracy of effort estimation in agile projects. Galorath and Evans (2006) argue that teams that effectively implement performance tracking and management practices are more likely to produce accurate effort estimates and achieve better project outcomes, such as on-time delivery and stakeholder satisfaction. Galorath and Evans (2006) highlight the necessity of continuous performance monitoring in ensuring alignment between effort estimates and actual project progress. By implementing performance tracking techniques and utilizing tools such as burndown charts and velocity measurements, teams can identify deviations from initial estimates, adjust their plans accordingly, and ultimately improve the overall estimation process.

MacDonell & Shepperd (2003) emphasizes the importance of using prior-phase effort records to inform effort estimations during software projects. By incorporating historical data, project managers can better understand the actual effort expended on similar tasks in the past, leading to more accurate estimations of the remaining work, which can be interpreted as the importance of recording estimations. Galorath & Evans (2006) emphasized a positive correlation between performance tracking and the level of quality in software projects. This means that projects that consistently measure and track their performance are more likely to achieve higher levels of quality. By identifying and addressing issues early in the development process, project teams can improve the overall quality of their software products. Srivastava, Trehan, Wagle, & Wang (2020) suggested high developer velocity enables organizations to deliver software products faster and with higher quality, allowing them to respond more effectively to changing market conditions and customer needs. This agility can translate into improved business performance and a competitive advantage.

To conclude, performance tracking and management play a critical role in the effort estimation process, providing valuable feedback on the accuracy of estimates and enabling continuous improvement. By employing performance tracking techniques, such as burndown charts and velocity tracking, teams can better understand their progress, identify deviations from estimates, and make necessary adjustments to ensure successful project delivery.

CHAPTER 4

RESEARCH METHODOLOGY

4.1 Research design

The research design employed in this study is a quantitative, cross-sectional survey-based approach to compare the man-day and story points effort estimation sizing measures in the software development process. This design allows for the collection of data from a large number of professionals at a single point in time, providing a snapshot of the current state of practice in effort estimation (Evans & Mathur, 2005).

By using a survey, insights and opinions from professionals with various levels of experience and backgrounds in software development are gathered. The main objective of this research design is to determine the differences and similarities between the two effort estimation methods in the context of experience, flexibility, complexity, teamwork, and performance tracking, as well as their possible subfactors, in light of excessive literature review.

To achieve this, a comprehensive two-part survey instrument has been designed. The first part consists of demographic questions. These demographic variables, including age, gender, experience, position, education, and sizing measure are considered to provide context for the professionals' responses and to control for potential confounding variables.

The second part focuses on the main factors (Experience, Flexibility, Complexity, Teamwork and Performance Tracking) and their possible corresponding subfactors.

Each factor is represented by 9 to 15 questions, which participants are asked to rate on a Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). This scale allows for the quantification of participants' attitudes and perceptions about the different aspects of man-day and story points estimation methods.

To analyze the data and test the research hypotheses, descriptive statistics, reliability analysis, factor analysis, and independent samples t-tests are used as statistical analysis methods.

4.2 Participants and sampling

The participants in this study were professionals working in the software development industry, who had experience using either the man-day or story points effort estimation sizing measures. They represented a diverse range of occupations within the field, including developers, analysts, project managers, managers, and team leads. This diversity ensured a comprehensive representation of perspectives and experiences in the study.

A non-probability, purposive sampling technique was employed to select professionals who were knowledgeable and experienced in using either of the estimation methods under investigation. This sampling approach allowed us to target individuals who could provide valuable insights into the subject matter.

The survey was distributed through various online channels, including professional chat groups and social media groups that catered to the specific occupations of developers, analysts, project managers, managers, and team leading professionals were informed about the purpose of the study, the anonymity of their responses, and the voluntary nature of their participation.

By gathering a diverse and representative sample of professionals working in the software development industry across various occupations, this study aims to provide a comprehensive understanding of the differences and similarities between the man-day and story points effort estimation methods in the context of experience, flexibility, complexity, teamwork, and performance tracking, as well as their possible subfactors.

4.3 Data collection

The demographic information collected from the professionals included age, position, education, sex, and the sizing measure they used in their professional practice.

The total number of professionals in the study was 530, after the exclusion of 13 incoherent responses. Out of these, 268 professionals had experience using the man-day estimation method, while 262 professionals were experienced in using the story points estimation method. Data collection took place from January 2023 to March 2023.

4.4 Hypotheses

The main objective of this study is to compare the man-day and story points effort estimation sizing measures in the context of experience, flexibility, complexity, teamwork, and performance tracking. Based on the literature review, following hypotheses are developed:

H_1: There is a significant difference in experience level of software professionals using man-day and story points estimation methods.

H_2: There is significant difference in the perceived flexibility of software development processes using man-day and story points estimation methods.

H_3: There is significant difference in the perceived complexity management capabilities of man-day and story points estimation methods.

H4: There is significant difference in the perceived teamwork importance between man-day and story points estimation methods.

H5: There is significant difference on importance of performance tracking between man-day and story points estimation methods.

4.5 Data analysis

The data analysis section of the study focuses on evaluating the collected data to draw meaningful conclusions and insights regarding the comparison of man-day and story points effort estimation sizing measures. The data analysis process comprises several stages, as described below:

Data Cleaning and Preparation: The initial stage of the data analysis process involves cleaning and preparing the collected data for further analysis. In this stage, 13 incoherent responses were identified and excluded from the dataset, resulting in a final sample size of 530 professionals.

Descriptive Analysis: Descriptive statistics were computed for the demographic variables, such as age, position, education, sex, and used sizing measure, to better understand the composition of the participant sample and control for potential confounding variables (Cohen, Manion, & Morrison, 2007).

Factor Analysis: The analysis conducted to identify possible subfactor for each main factors to form structure for the study (Rummel, 1988).

Reliability Analysis: Cronbach's alpha reliability analysis was conducted on the possible subfactors to ensure the internal consistency of the survey instrument (George & Mallery, 2018).

Independent Samples T-Test: To test the hypotheses and identify any significant differences between the man-day and story points estimation methods, independent samples t-tests were conducted. The t-test results will indicate whether any statistically significant differences exist between the groups for the main factors and their possible subfactors (Kim, 2015).

By following this structured data analysis process, the study aims to produce valid and reliable findings that contribute to the understanding of man-day and story points estimation methods in the software development process.

CHAPTER 5

RESULTS AND FINDINGS

In this chapter, a detailed analysis of the findings from the survey conducted with software professionals using either man-day or story points estimation methods are presented. The purpose of this analysis is to identify and examine any significant differences between these two groups concerning the main factors of experience, flexibility, complexity, teamwork, and performance tracking, as well as their possible corresponding subfactors.

The demographic characteristics of the participant sample are described, and potential confounding variables are controlled for. Factor analysis result are shown to identify underlying relationship between survey items. The results of the reliability analysis, which was performed to ensure the internal consistency of the survey instrument, are discussed. Finally, independent samples t-tests are used to test the research hypotheses and identify any significant differences between the man-day and story points estimation methods, as well as the demographic variables, such as age, position, experience, and gender.

By the end of this chapter, valuable insights into the impact of man-day and story points estimation methods on software development projects are aimed to be provided, offering guidance for practitioners and researchers alike in the field of effort estimation.

5.1 Demographic characteristics of the sample

In this section, the demographic characteristics of the survey participants are presented to provide a comprehensive understanding of the sample's composition.

These characteristics include age, position, education, gender, and the sizing measure used by the professionals.

Age: The age distribution of the participants is described, highlighting the age groups represented in the sample.

Position: The professional positions held by the participants, such as developers, analysts, project managers, team leads, or upper-level managers, are detailed.

Education: The education levels of the participants, including high school, vocational school, bachelor’s, master’s or doctorate degree are presented.

Gender: The gender distribution of the sample is provided.

Used Sizing Measure: The proportion of participants using man-day and story points estimation methods are reported, providing an overview of the sample's preference for each technique.

By exploring the demographic characteristics of the sample, this section will set the stage for a more in-depth analysis of the research hypotheses in the subsequent sections of this chapter.

5.1.1 Age

The participant age distribution was separated into five distinct age categories, which offers a clearer perspective on the age structure of the sample. The table below displays the count and corresponding percentages for each age category.

Table 1. Age Distribution of the Sample

Age Group	# of Participants	Percentage (%)
18-23	55	10.4%
24-30	196	37.0%
31-35	183	34.5%
36-45	70	13.2%
46+	26	4.9%

Table 1 showcases the participant age distribution, separated into five specific age categories, facilitating a clearer grasp of the sample's age characteristics. As shown in the Table 1, the largest group of participants (37%) falls into the 24-30 age range, closely followed by the 31-35 age range at 34.5%. Age groups 18-23 and 36-45 have lower representation rates, at 10.4% and 13.2% respectively, while the smallest percentage is seen in the 46+ age range, making up just 4.9% of all participants.

This distribution highlights that most participants are comparatively young, mainly ranging from 24 to 35 years old. This finding corresponds to the broader demographic trend observed in the software development sector, in which a sizable part of the workforce is commonly made up of young professionals.

5.1.2 Experience

The work experience in current position of the survey respondents was divided into five categories to offer a clearer understanding of the sample's varying experience levels. The table below presents the count and corresponding percentages for each experience category.

Table 2. Experience Distribution of the Sample

Experience	# of Participants	Percentage (%)
0-2 years	60	11.3
3-5 years	213	40.2
6-8 years	144	27.2
9-12 years	77	14.5
12+ years	36	6.8

Table 2 reveals that the majority of participants (40.2%) possess 3-5 years of experience. Those with 6-8 years of experience comprise 27.2%. Participants with 0-2 years of experience account for 11.3% of the sample. Individuals with 9-12 years of experience represent 14.5%. The smallest group (6.8%) has 12 or more years of experience.

This distribution of experience within the sample indicates that most survey respondents have between 3 and 8 years of professional experience. This finding aligns with the common demographic trend in the software development industry, where a significant portion of the workforce consists of professionals in the intermediate stages of their careers.

5.1.3 Position

The job titles held by the survey respondents were divided into five categories to offer a clearer understanding of the sample's role distribution. The count and corresponding percentages for each role are displayed in the table below.

Table 3. Position Distribution of the Sample

Position	# of Participants	Percentage (%)
Software Developers	205	38.7%
Analysts	237	44.7%
Project Managers	33	6.2%
Team Leads	29	5.5%
Managers	26	4.9%

Table 3 indicates that the majority of participants were Analysts (44.7%) and Software Developers (38.7%). Project Managers, Team Leads, and Managers constituted smaller percentages, at 6.2%, 5.5%, and 4.9% respectively.

The role distribution within the sample suggests that the survey mainly targeted individuals directly engaged in the software development process, such as Software Developers and Analysts. Participants in managerial and leadership positions, like Project Managers, Team Leads, and Managers, formed a smaller proportion of the sample.

5.1.4 Education

The education levels of the survey participants were categorized into five groups to provide a better understanding of the sample's educational background. The number of participants and their relevant percentages for each education level are presented in the table below.

Table 4. Education Level Distribution of the Sample

Education Level	# of Participants	Percentage (%)
High School Degree	3	0.6%
Vocational School Degree	11	2.1%
Bachelor's Degree	475	89.6%
Master's Degree	37	7.0%
Doctorate Degree	4	0.8%

Table 4 shows that 89.6% of participants have a bachelor's degree. Master's degree holders are 7.0%, and vocational school graduates make up 2.1%. High school degree holders make up 0.6% of the participants. Doctorate degree holders represent a slightly higher proportion at 0.8%.

The education distribution of the sample indicates that most survey participants had completed higher education in the form of an undergraduate degree. This reflects the general trend in the software development field where a bachelor's degree is often the minimum requirement for employment.

5.1.5 Gender

The gender composition of the survey participants was separated into male and female to provide better insight into the sample's gender demographics. The table below presents the number of participants and their corresponding percentages for each gender.

Table 5. Gender Distribution of the Sample

Gender	Number of Participants	Percentage (%)
Male	354	66.8%
Female	176	33.2%

According to Table 5, 66.8% of the sample were male. Female made up 33.2% of the total participants of the sample. The gender distribution of the sample reflects the general trend in software development, where the workforce has historically been predominantly male. However, including both male and female views in the analysis helps avoid bias towards a single gender.

5.1.6 Sizing measure

The sizing measure used by the survey participants was categorized into two groups, man-day and story points. The number of participants and their relevant percentages for each sizing measure are presented in the table below.

Table 6. Sizing Measure Distribution of the Sample

Sizing Measure	# of Participants	Percentage (%)
Man-Day	268	50.6
Story points	262	49.4

As seen in the Table 6, the participants were almost equally divided between the two estimation methods. Man-day estimation was used by 50.6% of the sample, while story points estimation was used by 49.4% of the participants. This balanced representation ensures that the analysis will provide a fair comparison of the two estimation methods, considering the perspectives of professionals who use each method.

5.2 Factor analysis

Following the assessment of demographic structure of the survey participants, the next essential step in our analytical process is to examine the underlying structure of the survey through factor analysis. In this section, the factorability of the questionnaire items is assessed to determine the existence of coherent subsets that reflect different aspects of the construct under study. Factor analysis enables us to understand the patterns of relationships among the items and to explain the variance among the observed variables in terms of a smaller number of unobserved variables, known as 'factors'. This powerful statistical method helps to simplify the data, reduces the risk of multicollinearity in subsequent analyses, and facilitates more accurate and meaningful interpretation of the results. The pattern matrix resulting from the factor analysis, which illustrates the factor loadings of each item, can be found in the subsequent tables except loading values between -0.30 and 0.30.

Upon examination of the factor analysis results, with loading cumulative score of 59.2% and KMO measure of sampling accuracy of .668, it became apparent that certain items were not exclusively loading onto a single factor. These cross-loading items, in effect, exhibited strong associations with more than one factor, indicating a potential overlap in the constructs they measure. According to Howard (2006), it would be preferable to eliminate the cross-loading items, particularly if there are sufficient strong loaders in the factor loading dimensions and the cross-loading items have factor loading scores above 0.3. In result, survey items Experience - 4, Experience – 5, Experience - 6 are deducted from the study.

Table 7. Pattern Matrix of Cross Loading Items

Item	Component 1	Component 2	Component 3
Experience – 1	.769		
Experience – 2	.770		
Experience – 3	.816		
Experience – 4	.448	.342	
Experience – 5	.465	.425	.338
Experience – 6	.419	.462	.314
Experience – 7		.649	
Experience – 8		.807	
Experience – 9		.724	
Experience – 10			-.623
Experience – 11			-.684
Experience – 12			-.679

Following the careful consideration and subsequent removal of the cross-loading items, a new factor analysis was performed to examine the robustness of the constructs of the study, with result of loading cumulative score of 70.1% and KMO measure of sampling accuracy of .783. In the factor analysis conducted for this research survey, three main components were extracted from the set of experience-related questions. These components appear to represent distinct facets of the broader concept of experience. The first component comprised of questions labeled Experience-1, Experience-2, and Experience-3 and primarily represented the aspect of self-assessment subfactor. It seems to capture respondents' self-perceptions or judgments about their own skills, abilities, and experiences. The second component, containing questions Experience-7, Experience-8, and Experience-9, reflected the respondents' number of estimated projects subfactor, serving as a more quantifiable, objective measure of their experience. Finally, the third component, encompassing

Experience-10, Experience-11, and Experience-12, revolved around the theme of implementation experience subfactor, capturing hands-on, practical experience or knowledge gained from directly applying concepts or theories in real-world contexts. These three components collectively provide a multifaceted understanding of the experience construct as it applies to the survey participants. Detailed factor loadings for each construct can be found in Table 7.

Table 8. Pattern Matrix of Survey Items - 1

Item	Component 1	Component 2	Component 3
Experience-1	.791		
Experience-2	.819		
Experience-3	.818		
Experience-7		.725	
Experience-8		.802	
Experience-9		.763	
Experience-10			-.817
Experience-11			-.792
Experience-12			-.817

Following the careful consideration and subsequent removal of the cross-loading items, a new factor analysis was performed to examine the robustness of the constructs of the study, with result of loading cumulative score of 70.1% and KMO measure of sampling accuracy of .783. In the factor analysis conducted for this research survey, three main components were extracted from the set of experience-related questions. These components appear to represent distinct facets of the broader concept of experience. The first component comprised of questions labeled Experience-1, Experience-2, and Experience-3 and primarily represented the aspect of self-assessment subfactor. It seems to capture respondents' self-perceptions or judgments about their own skills, abilities, and experiences. The second component, containing questions Experience-7, Experience-8, and Experience-9, reflected the respondents' number of estimated projects subfactor, serving as a more quantifiable, objective measure of their experience. Finally, the third component, encompassing Experience-10, Experience-11, and Experience-12, revolved around the theme of implementation experience subfactor, capturing hands-on, practical experience or knowledge gained from directly applying concepts or theories in real-world contexts. These three components collectively provide a multifaceted understanding of the experience construct as it applies to the survey participants. Detailed factor loadings for each construct can be found in Table 8.

Table 9. Pattern Matrix of Survey Items - 2

Item	Component 4	Component 5	Component 6
Flexibility-1	-.805		
Flexibility-2	-.762		
Flexibility-3	-.827		
Flexibility-4		.824	
Flexibility-5		.718	
Flexibility-6		.767	
Flexibility-7			.685
Flexibility-8			.815
Flexibility-9			.837

The outcomes at Table 9 affirmed the integrity of the constructs. The factor analysis of the questions concerning flexibility resulted in the formation of three distinct components, each representing a unique dimension of flexibility. The first component consisted of questions Flexibility-1, Flexibility-2, and Flexibility-3 and was centered around the theme of flexibility in process subfactor. This dimension captures the adaptability and versatility of methods in adjusting their estimates in response to varying circumstances. The second component, including questions

Flexibility-4, Flexibility-5, and Flexibility-6 focused on adaptability to changes subfactor. It appears to assess the capacity to modify or transform estimates to effectively deal with new or changing environments, conditions or situations. The third component, with questions Flexibility-7, Flexibility-8, and Flexibility-9, revolved around the theme of robustness subfactor, capturing the ability to maintain estimation accuracy in the face of disturbances or adversities. Collectively, these components provide a comprehensive picture of the concept of flexibility as it is captured by this survey.

Table 10. Pattern Matrix of Survey Items - 3

Item	Component 7	Component 8	Component 9
Complexity-1	-.823		
Complexity-2	-.776		
Complexity-3	-.806		
Complexity-4		.804	
Complexity-5		.824	
Complexity-6		.790	
Complexity-7			.727
Complexity-8			.807
Complexity-9			.847

The complexity-related questions in the research survey were parsed into three unique components via factor analysis, each addressing a different facet of complexity. The first component comprised questions Complexity-1, Complexity-2, and Complexity-3 and revolved around the theme of dependencies on other systems subfactor. This dimension gauges the interdependence of various systems, and the complexity that arises from managing these dependencies while making estimations. The second component contained questions Complexity-4, Complexity-5, and Complexity-6 and focused on backlog item complexity subfactor, representing the difficulty associated with individual tasks or items in a project's backlog. The final component, consisting of questions Complexity-7, Complexity-8, and Complexity-9, captured the theme of task uncertainty subfactor. This component measures the level of ambiguity and unpredictability associated with tasks, thereby assessing another key aspect of complexity. Together, these components provide a well-rounded understanding of the multifaceted nature of complexity within the survey's context.

Table 11. Pattern Matrix of Survey Items - 4

Item	Component 10	Component 11	Component 12
Teamwork-1	-.805		
Teamwork-2	-.796		
Teamwork-3	-.636		
Teamwork-4		.770	
Teamwork-5		.763	
Teamwork-6		.757	
Teamwork-7			.739
Teamwork-8			.742
Teamwork-9			.740

The factor analysis of the teamwork-related questions led to the construction of five distinct components, each addressing a unique aspect of team dynamics. The first component, encompassing questions Teamwork-1, Teamwork-2, and Teamwork-3, centered on the theme of trust on each other subfactor. This dimension captures the level of trust and mutual reliance among team members in effort estimation. The second component, derived from questions Teamwork-4, Teamwork-5, and Teamwork-6, represented participation subfactor. It measures the degree of active involvement and contribution of each team member to the group tasks about effort estimation. The third component, consisting of questions Teamwork-7, Teamwork-8, and Teamwork-9, related to team member competency subfactor, assessing the skill level of individual team members while making estimations.

Table 12. Pattern Matrix of Survey Items - 5

Item	Component 13	Component 14
Teamwork-10	-.819	
Teamwork-11	-.827	
Teamwork-12	-.765	
Teamwork-13		-.817
Teamwork-14		-.872
Teamwork-15		-.833

The fourth component, which comprised questions Teamwork-10, Teamwork-11, and Teamwork-12, focused on commonalities subfactor. It evaluates the shared rituals within a team that facilitate effective collaboration while making effort estimation. Lastly, the fifth component, consisting of questions Teamwork-13, Teamwork-14, and Teamwork-15, was themed around learning from experience subfactor. This component gauges the team's ability to learn and adapt based on past experiences or challenges. Together, these components offer a comprehensive evaluation of the team dynamics at play within the survey context.

Table 13. Pattern Matrix of Survey Items - 6

Item	Component	Component	Component
Performance Tracking-1	-.901		
Performance Tracking-2	-.893		
Performance Tracking-3	-.883		
Performance Tracking-4		-.855	
Performance Tracking-5		-.881	
Performance Tracking-6		-.894	
Performance Tracking-7			.787
Performance Tracking-8			.791
Performance Tracking-9			.823

In the factor analysis conducted on the performance tracking related questions, three distinctive components were identified, each encapsulating a different facet of performance tracking. The first component was constructed with questions Performance Tracking-1, Performance Tracking-2, and Performance Tracking-3, and centered on the concept of velocity subfactor. This dimension measures the speed at which tasks or projects are completed, providing a dynamic assessment of performance over time. The second component, consisting of questions Performance Tracking-4, Performance Tracking-5, and Performance Tracking-6, was themed around the level of quality subfactor. This offers an evaluation of the standard or grade of the completed work, contributing a measure of performance quality to the analysis. The third component, built from questions Performance Tracking-7, Performance Tracking-8, and Performance Tracking-9, was concerned with recording estimations subfactor. This component related to importance and ease in recording estimations. Together, these components offer a comprehensive framework for understanding and assessing various dimensions of performance tracking in the research context.

5.3 Reliability analysis

Names of subfactors is given in previous section in light of literature review. Also, relationship between subfactors and main factors in literature is mentioned in Chapter 2. Before proceeding with the analysis of the research hypotheses, it is very important to ensure the internal consistency of the survey. Before proceeding with the analysis of the research hypotheses, it is very important to ensure the internal consistency of the survey. In this section, the reliability analysis on the determent sub-factors within the scope of the study is checked. The purpose of this analysis is to determine to what extent the questionnaire items measuring each determined subfactor are consistent with each other and thus to provide confidence in the results obtained from the questionnaire responses. Cronbach's alpha, a widely used internal consistency measure, is used to evaluate the reliability of the determined subfactors. Reliability analysis results can be seen table below.

Table 14. Cronbach's Alpha Values of Items of Determined Subfactors for each Main Factor

Main Factor	Determined Subfactor	Cronbach's
Experience	Implementation Experience	0.750
	# of Estimated Project	0,721
	Self-Assessment	0,799
Flexibility	Flexibility in Process	0,748
	Adaptability to the Changes	0,733
	Robustness	0,730
Teamwork	Trust on Each Other	0,728
	Commonalities	0,786
	Learn from Experience	0,826
	Team Member Competency	0,704
	Participation	0,749
Complexity	Dependencies on Other Systems	0,750
	Backlog Item Complexity	0757
	Uncertainty of the Task	0,748
Performance Tracking	Recording Estimations	0,750
	Level of Quality	0,865
	Velocity	0,888

The results of the reliability analysis, as displayed in Table 14, indicate that all Cronbach's alpha values are greater than 0.7. This demonstrates a high level of internal consistency among the determined subfactors. A Cronbach's alpha value above 0.7 is generally considered acceptable and indicates that the survey items are reliable and measuring the intended constructs effectively (Cortina, 1993).

The high reliability of the determined subfactors provides confidence in the data obtained from the survey responses and ensures that the analysis and interpretation of research hypotheses is based on consistent and reliable information. As a result, the findings from this study are more likely to be valid and can be used to draw meaningful conclusions about comparing man-day and story points estimation methods in the context of experience, flexibility, complexity, teamwork, and performance tracking.

5.4 Independent samples t-test

Upon completion of the multiple reliability analysis, the research will proceed to evaluate the differences between the man-day and story points estimation methods using independent samples t-tests. This statistical technique enables the comparison of the means of two distinct groups to determine if a significant difference exists between them for a specific variable. In the context of this investigation, the independent samples t-test is applied to test the research hypotheses concerning the disparities in experience, flexibility, complexity, teamwork, and performance tracking between the two groups. The outcomes of the independent samples t-tests will offer valuable insights into the contrasts between man-day and story points estimation methods, aiding in the identification of any significant differences in the main factors and their determined subfactors.

5.4.1 Comparison of man-day and story points estimation methods in main factors

In order to explore and evaluate the differences in the main factors between the man-day and story points estimation methods, the study employed the use of independent samples t-tests. These tests were conducted with the specific objective of comparing the mean values of the key influencing factors between these two different estimation approaches. To aid in the interpretation and understanding of our results, the study has provided the significance levels for each factor, which are calculated based on a 2-tailed test.

Table 15. Independent Samples T-tests of Main Factors

Main Factor	Significance Level (2-tailed)
Experience	0,220
Flexibility	0,409
Teamwork	0,215
Complexity	0,103
Performance Tracking	0,098

The t-test result for experience shows a significance level of 0.22, which is greater than the threshold of 0.05. This indicates that there is no significant difference in the experience level of software professionals using man-day and story points estimation methods. Hence, H_1 is rejected.

The t-test result for flexibility yields a significance level of 0.409, which is above the threshold of 0.05. This suggests that there is no significant difference in the perceived flexibility of software development processes using man-day and story points estimation methods. Consequently, H_2 is rejected.

The t-test result for teamwork presents a significance level of 0.215, which is greater than the threshold of 0.05. This finding implies that there is no significant difference in the perceived teamwork importance between man-day and story points estimation methods. Thus, H_3 is rejected.

The t-test result for complexity displays a significance level of 0.103, which is above the threshold of 0.05. This result indicates that there is no significant difference in the perceived complexity management capabilities of man-day and story points estimation methods. Thus, H_4 is rejected.

The t-test result for performance tracking shows a significance level of 0.098, which is greater than the threshold of 0.05. This finding suggests that there is no significant difference in the importance of performance tracking between man-day and story points estimation methods. As a result, H_5 is rejected.

Based on the t-test results, it can be concluded that there are no significant differences between man-day and story points estimation methods in terms of experience, flexibility, teamwork, complexity, and performance tracking.

5.4.2 Comparison of man-day and story points estimation methods in determined subfactors

The independent samples t-tests were conducted to compare the means of the determined subfactors between the man-day and story points estimation methods, determined subfactor items were averaged. The significance levels (2-tailed) for each subfactor are provided in table below.

Table 16. Independent Samples T-tests of Determined Subfactors

Main Factor	Determined Subfactor	Significance Level (2-
Experience	Implementation Experience	0,383
	Number of Estimated Projects	0,095
	Self-Assessment	0,404
Flexibility	Flexibility in Process	0,179
	Adaptability to the Changes	0,710
	Robustness	0,000
Teamwork	Trust on Each Other	0,154
	Commonalities	0,985
	Learn from Experience	0,717
	Team Member Competency	0,331
	Participation	0,159
Complexity	Dependencies on Other Systems	0,080
	Backlog Item Complexity	0,774
	Uncertainty of the Task	0,000
Performance Tracking	Recording Estimations	0,778
	Level of Quality	0,014
	Velocity	0,542

As can be seen on Table 16, there are significant differences between the man-day and story points estimation methods for the determined subfactors of Robustness and Uncertainty of the Task, as their values are 0, which is below the threshold of 0,05. On average, professionals who use the story points method scored 3,521 while professionals who use man-day are scoring 3,884 on Robustness. Moreover, on average, professionals who use the man-day method scored 3,501 while professionals who use story points are scoring 4,058 on Uncertainty of the

Task. Additionally, Level of Quality shows a significant difference with a value of 0,014, which is also below the threshold. While professionals who use man-day method are scoring 3,633 on average, professionals who use story points technique scored 3,924. For the remaining determined subfactors, the significance level being greater than 0.05 indicates that there is no significant difference between the two estimation methods.

CHAPTER 6

CONCLUSION

This chapter discusses the results obtained in the research, indicates potential similarities and differences of professionals' tendencies between man-day and story point approach, explains the limitations faced and finally, offers suggestions for future research.

6.1 Discussion of the results

In this study, a total of 5 hypotheses were formulated to examine the relationships between main factors and their determined subfactors and to identify potential differences between software professionals who use man-day or story points. The findings were supported by empirical evidence derived from independent samples t-tests with using SPSS data analysis tool.

Hypotheses H_1, H_2, H_3, H_4, H_5 focused on the comparison of the main concepts between professionals using man-day and story points estimation methods. The interpretation of the hypotheses and findings aimed to determine if there are significant differences between man-day and story points estimation methods in terms of experience, flexibility, teamwork, complexity, and performance tracking. Findings show that H_1, H_2, H_3, H_4, H_5 are rejected.

The first hypothesis suggested a significant difference in the experience level of software professionals using man-day and story points estimation methods, assuming as professionals get more experienced, they tend to choose either one of the estimation methods. However, our findings do not support the hypothesis, indicating that both novice and experienced software professionals employ both methods. In addition, no difference is found in independent samples t-test for subfactors that related to experience in this study. This is dissonant with findings from previous research, suggesting that the choice of estimation technique is significantly influenced by the experience level (Prayitno et al. 2020). The difference between our results and the literature may stem from the nature of our sample. In total, survey participants who has 0-5 years of experience takes up to 51% of sample in our study. The weight of less experienced survey participants may influence the study result.

The second hypothesis suggested a significant difference in the perceived flexibility between the man-day and story points estimation methods. With assumption of story points technique's advantage of flexible estimation unit model (Fibonacci series), H_2 is developed. According to results, it is concluded that either method that study tested is indifferent from each other in terms of perceived flexibility. Our result contradicts with suggestions of Anastasiya & Viktoria (2020), since they indicate that story points method is more flexible compared to man-day approach. In addition, surprisingly, independent samples t-test for subfactors indicates that professionals who use man-day approach believe that their technique is more robust than story points users. The time series that man-day method utilize may be the cause of this unexpected result, since it perceived less abstract than story points method. In the man-day approach, the effort is assigned in a precise time, for instance 2 days or 3 weeks. In contrast, in story points method, effort is assigned according to Fibonacci series that represent the value of a task and values may differ team to team. This makes man-day method to be understood easily by every stakeholder and project the costs better than story points method.

The third hypothesis suggested a significant difference in the perceived complexity management capabilities between the man-day and story points estimation methods. Presuming story points method is more preferable compared to man-day method. However, results of the study indicates that there is no significant difference between man-day and story points approach. Hence, H_3 is rejected. In contrast to study result. Sree et al. (2019) suggests that use case points, story points and functional points methods are more preferable for effort estimation in the early stage of software development compared to other models in case of development complexity. The lack of significant differences may be attributed to the similar fundamental principles underlying both methods. They aim to estimate the effort required for software development tasks, considering task complexity. Still given no significant differences on complexity aspect of both method, story points method users believe that when task possess uncertainties their method is more favorable compared to man-day method.

Since, story points method is applied variously in different development teams, it requires consensus on unit identification by team members. This condition necessitates more collaboration compared to man-day approach which only estimates individuals' work effort. By considering this, H_4 is proposed, in favor of story points method. However, according to study result, it can be seen that there was no difference between man-day and story points approach in case of perceived teamwork importance. In addition, no significant differences are found in subfactors that related to teamwork either. This result aligns with the study of Moe & Dingsøyr & Dybå (2010) that indicates regardless of the estimation method used, teamwork is considered equally essential in software development projects. Both findings reflect the inherent team-oriented nature of software development work and the essential role of collaboration and coordination, regardless of the specific effort estimation method employed.

The choice of an accurate measure is crucial because different estimation techniques behave differently based on the accuracy measures used. This suggests that the choice of estimation technique may consider the need for accurate performance tracking to evaluate the effectiveness of the chosen technique. In this manner, the study aimed to investigate which method is more favorable in term of performance tracking. In order to investigate comparison, H_5 is proposed, in favor of man-day estimation method. Man-day estimation method assign actual timely effort to complete task while story points use abstract Fibonacci series model. With this nature, man-day approach is expected to be more precise to monitor individuals' performance. However, according to our study results, it can be concluded that there is no difference between man-day and story points method in terms of importance of performance tracking. Idri et al. (2017) emphasize the significance of choosing a reliable estimation accuracy measure. Our result align with this research's result, in terms of comparing man-day and story points techniques in context of performance tracking. Both techniques are assessed equally by the participants. On the other hand, surprisingly, level of quality which is a performance tracking subfactor found to be more favorable by story points users compared to man-day method users. It is assumable since while assigning in story points method, practitioners also focus on the complexity and quality rather than merely the time it would take to complete it.

In conclusion, the results suggest that man-day and story points estimation methods are not significantly different in terms of experience level of software professionals, perceived flexibility of software development processes, perceived teamwork importance, perceived complexity management capabilities and the importance of performance tracking.

6.2 Implications of the research

This study has significant implications for both academia and the software industry, particularly in relation to effort estimation techniques, man-day and story points methods. Firstly, it challenges the previously held belief that experience level significantly influences the selection of estimation methods, implying that both novice and experienced software professionals employ both methods indiscriminately. This understanding could be fundamental in further research and in professional practices, as it brings to light the importance of individual and team preferences, knowledge, and contextual factors in the choice of estimation techniques. Secondly, the study results provide a nuanced perspective on the perceived flexibility, complexity management, teamwork importance and performance tracking between man-day and story points methods, an area that was previously believed to be distinctive. The study findings emphasize that professionals do not perceive one method as inherently superior to the other in these aspects, offering critical insights to the organizations and practitioners. This suggests that the focus should be on how the techniques are utilized and tailored to meet unique project requirements, rather than which technique is theoretically superior.

6.3 Limitations and future research suggestions

The present study has provided valuable insights into the differences between software professionals in various aspects. However, there are still several areas that can be explored in future research to further our understanding of the software development field.

One of the limitations of this study pertains to the data collection method concerning the sizing measures used by participants. In our survey, participants were asked to provide information solely about their current sizing measures, either man-day or story points. This approach might have limited our understanding of the full range of sizing measures that participants may have experience with or the potential influences of their past practices on their current perspectives. Furthermore, it does not account for possible variations in the interpretation and application of these methods across different contexts. In future research, it would be beneficial to gather information about all the sizing measures that the participants have used in their careers. This could provide a more comprehensive picture of their experiences and potentially reveal more nuanced insights into the relative strengths and weaknesses of different estimation methods.

Secondly, the present study focused on specific factors, such as experience, flexibility, complexity, teamwork, and performance tracking. Future research could explore additional factors that may influence software professionals' work, such as communication styles, leadership, and work-life balance. Besides that, longitudinal studies could be conducted to investigate how software professionals' perspectives on main and determined subfactors evolve over time. This would allow researchers to better understand the factors that contribute to the preferences of estimation methods. Also, future studies may benefit from incorporating qualitative research methods, such as interviews or focus groups, to obtain more in-depth insights into software professionals' experiences and perspectives. This could help uncover the underlying reasons for observed differences or similarities in various aspects of the study.

APPENDIX

SURVEY QUESTIONS

Section 1

1. Yazılım geliştirme alanında kaç yıl deneyiminiz var?
   (How many years of experience do you have in the field of software development?)
2. Kaç yaşınızdasınız?
   (How old are you?)
3. Pozisyonunuz?
   (What is your position?)
4. Cinsiyetiniz?
   (What is your gender)
5. Mezuniyet durumunuz?
   (What is your graduation status)
6. Kullandığınız eforlama ölçütü hangisidir?
   (Which effort estimation measure do you use?)

Section 2

1. Kullanılan yazılım diline hakimiyetin eforlama verirken önemli olduğunu düşünüyorum.
   (I believe that mastery of the programming language used is important when giving effort estimation.)

2. Kullanılan yazılım dilinde deneyimli olduğumu düşünüyorum.

   (I consider myself experienced in the programming language that I use.)

3. Kullanılan eforlama ölçütünde deneyimli olduğumu düşünüyorum.

   (I consider myself experienced in the effort estimation measure that I use.)

4. Yazılım geliştirme alanında deneyimli bir çalışan olduğumu düşünüyorum.

   (I consider myself an experienced employee in the field of software development.)

5. Yazılım geliştirme alanında deneyim yılının önemli bir faktör olduğunu düşünüyorum.

   (I think that years of experience in software development is an important factor.)

6. Yazılım geliştirmede eforlama ölçütünü kullanma deneyiminin önemli bir faktör olduğunu düşünüyorum.

   (I think that experience in using the effort estimation measure in software development is an important factor.)

7. Geçmişte eforlama ölçütü ile yapılan proje sayısının doğru eforla yapılmasında önemli bir ölçüt olduğunu düşünüyorum.

   (I think that the number of projects done with the effort estimation measure in the past is an important criterion for accurate effort estimation.)

8. Kullanılan eforlama ölçütünde tecrübelenmek için çok sayıda proje tahminleme yapılması gerektiğini düşünüyorum.

   (I think that many project estimations need to be made to gain experience in the effort estimation measure that I use.)

9. Çok sayıda projede eforlama yapmış takım üyelerinin daha deneyimli olduğunu düşünüyorum.

(I think team members who have made effort estimations in many projects are more experienced.)

10. Kullanmakta olduğum eforlama ölçütüne ve efor vermeye hakim olduğumu düşünüyorum.

    (I think I am proficient in the effort estimation measure I am using and in estimating effort.)

11. Kullanmakta olduğum yazılım dillerine hakim olduğumu düşünüyorum.

    (I think I am proficient in the programming languages I am using.)

12. Yazılım geliştirme süreçlerine hakim olduğumu düşünüyorum.

    (I think I am proficient in software development processes.)

13. Yazılım geliştirme sektöründe deneyimli bir bireyim.

    (I am an experienced individual in the software development industry.)

14. Kullanılan eforlama ölçütünün eforlama sürecinde esneklik sağladığını düşünüyorum.

    (I think the effort estimation measure that I use provides flexibility in the effort estimation process.)

15. Kullanılan eforlama ölçütünün süreçteki değişikliklere uyumlu olduğunu düşünüyorum.

    (I think the effort estimation measure that I use is adaptable to changes in the process.)

16. Gereksinimlerde değişiklik olma durumunda kullanılan eforlama ölçütünün esneklik sağladığını düşünüyorum.

    (I think the effort estimation measure that I use provides flexibility in case of changes in requirements.)

17. Eforlama ölçütünün tekrar planlama gerektiren durumlarda kolaylık yarattığını düşünüyorum.
    (I think the effort estimation measure that I use creates ease in situations requiring replanning.)
18. Planlanan eforlamayı rahatça değiştirebildiğimi düşünüyorum.
    (I think I can easily change the planned effort estimation.)
19. Kullanılan eforlama ölçütünün gereksinimlerde değişiklik olması durumunda uyumlu olduğunu düşünüyorum.
    (I think the effort estimation measure that I use is compatible in case of changes in requirements.)
20. Eforlamada sonradan yapılan değişikliklerin yapılan işlerde sıkıntı yaratmadığını düşünüyorum.
    (I think changes made later in effort estimation do not cause problems in the work done.)
21. Eforlamada yapılan değişikliklerin teslim tarihine etkisinin az olduğunu düşünüyorum.
    (I think the impact of changes made in effort estimation on the delivery date is minimal.)
22. Eforlama ölçütünün planlamadaki değişikliklere karşı dirençli bir yapıda olduğunu düşünüyorum.
    (I think the effort estimation measure is resistant to changes in planning.)
23. Kullandığım eforlama ölçütü esnek yapıdadır.
    (The effort estimation measure that I use is flexible.)
24. Ekip üyelerinin diğer üyelerin eforlama tahminlerine güvendiğini düşünüyorum.

(I think team members trust the effort estimations of other members.)

25. Ekip üyelerinin diğer üyelerin eforlama yorumlarına güvendiğini düşünüyorum.

    (I think team members trust the effort estimation comments of other members.)

26. Eforlama yaparken ekip içerisinde sözüne daha fazla güvenilen kişiler olduğunu düşünüyorum.

    (I think there are people whose words are trusted more when making effort estimations within the team.)

27. Ekip üyelerinin eforlama ölçütünü açıkça anladığını düşünüyorum.

    (I think team members clearly understand the effort estimation measure.)

28. Ekip üyelerinin eforlama süreçlerine hakim olduklarını düşünüyorum.

    (I think team members are proficient in effort estimation processes.)

29. Ekip üyelerinin eforlama için belirlenen ritüelleri takip ettiğini düşünüyorum.

    (I think team members follow the rituals for effort estimation.)

30. Ekip olarak geçmişte yapılan eforlamaların değerlendirmesini iyi yaptığımızı düşünüyorum.

    (I think as a team we evaluate past effort estimations well.)

31. Ekip olarak geçmişte yapılan eforlamaların doğruluk derecesini iyi takip ettiğimizi düşünüyorum.

    (I think as a team, we monitor the accuracy of past effort estimations well.)

32. Ekip olarak geçmişte yapılan hatalı eforlamaların nedenleri hakkında değerlendirme yaptığımızı düşünüyorum.

    (I think as a team, we evaluate the reasons for incorrect effort estimations made in the past.)

33. Takım üyelerinin yeterliliklerine güveniyorum.

(I trust the competencies of team members.)

34. Takım üyelerinin eforlama ölçütü hakkında net bilgi sahibi olduğunu düşünüyorum.

(I think team members have clear knowledge about the effort estimation measure.)

35. Takım üyelerinin doğru şekilde eforlama yaptığını düşünüyorum.

(I think team members make effort estimations correctly.)

36. Ekip üyelerinin eforlama sürecine dahil olduklarını düşünüyorum.

(I think team members are involved in the effort estimation process.)

37. Ekip üyelerinin diğer üyelerin eforlama tahminleri hakkında yorumda bulunduklarını düşünüyorum.

(I think team members comment on other members' effort estimations.)

38. Yapılacak işlerin eforlarına ortak şekilde karar verildiğini düşünüyorum.

(I think a collective decision is made on the effort of the tasks to be done.)

39. Takım çalışması eforlamada önemli olmuştur.

(Teamwork has been important in effort estimation process.)

40. Yapılacak geliştirmede diğer sistemlere bağımlılığının fazla olmasının eforlamayı zorlaştıracağını düşünüyorum.

(I think the high dependency on other systems in the development to be made will complicate the effort estimation.)

41. Kullanılan eforlama ölçütünün diğer sistemlere bağımlılık konusunda engel olmadığını düşünüyorum.

(I think the effort estimation measure that I use does not hinder the issue of dependency on other systems.)

42. Kullanılan eforlama ölçütün diğer sistemlere bağımlılığın kompleksliği arttırmadığını düşünüyorum.

    (I believe that the effort estimation measure that I use does not increase the complexity of dependence on other systems.)

43. Taskların kompleks olmasının eforlamayı zorlaştırdığını düşünüyorum.

    (I think the complexity of tasks makes effort estimation difficult.)

44. Eforlama ölçütünün kompleks projelerde gerçekleşen ve planlanan efor arasındaki farka etkisi olduğunu düşünüyorum.

    (I believe that the effort estimation measure has an effect on the difference between actual and planned effort in complex projects.)

45. Kullanılan efor ölçütünün kompleks taskları eforlamada kolaylık sağladığını düşünüyorum.

    (I think the effort estimation measure that I use makes it easy to estimate complex tasks' effort.)

46. Kullanılan eforlama ölçütünün belirsizliği fazla olan işlerde yararlı olduğunu düşünüyorum.

    (I believe the effort estimation measure that I use is useful in tasks with high uncertainty.)

47. Belirsizlik bulunan tasklarda eforlama ölçütünün, hata payının artmasına sebep olduğunu düşünüyorum.

    (I believe that the effort estimation measure that I use increases the margin of error in tasks with uncertainty.)

48. Belirsizlik içeren taskları eforlamada güçlük çektiğimi düşünüyorum.

    (I think I have difficulty in effort estimation in case of tasks with uncertainty.)

49. Kullanılan eforlama ölçütü kompleks değildir.

(The effort estimation measure that I use is not complex.)

50. Kullanılan eforlama ölçütünün eforları kaydetmede kullanışlı olduğunu düşünüyorum.

(I think the effort estimation measure used is useful in recording effort estimations.)

51. Efor tahminlemelerini kaydetmenin gelecek eforlamalar için yararlı bir pratik olduğunu düşünüyorum.

(I think recording effort estimations is a useful practice for future effort estimations.)

52. Kaydedilen eforların performans takibinde yararlı olduğunu düşünüyorum.

(I think the recorded efforts are useful in performance tracking.)

53. Kullanılan eforlama ölçütünün çıkan işin kalitesini ölçmede bir ölçüt olduğunu düşünüyorum.

(I think the effort estimation measure that I use is a criterion for measuring the quality of the resulting work.)

54. Kullanılan eforlama ölçütünün kaliteli çıktı oluşmasında bir faktör olduğunu düşünüyorum.

(I think the effort estimation measure that I use is a factor in creating quality output.)

55. Kalite takibi içi eforların kaydedilmesinin yararlı olduğunu düşünüyorum.

(I think it's beneficial to record efforts for quality tracking.)

56. Kullanılan eforlama ölçütü sayesinde takımın hız performansını ölçebileceğimizi düşünüyorum.

(I think we can measure the velocity of the team thanks to the effort estimation measure that I use.)

57. Kullanılan eforlama ölçütünün takım performansını takip etmek için iyi bir araç olduğunu düşünüyorum.

    (I think the effort estimation measure that I use is a good tool to track team performance.)

58. Kullanılan eforlama ölçütünü ekip performansını raporlamada kullandığımızı düşünüyorum.

    (I think we involve the effort estimation measure to report team performance.)

59. Kullanılan eforlama ölçütü performans takibi için yararlı olmuştur.

    (The effort estimation measure that I use has been useful for performance tracking.)

REFERENCES

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11).

Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, *12*(1), 106.

Anastasiya V, & Viktoria K. (2020, December 14). Story Points vs Hours: 4 Reasons to Estimate with Story Points. Retrieved from Rubygarage.org

Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press.

Beck, K. (2000). *Extreme programming explained: embrace change.* Addison-Wesley Professional.

Begel, A., & Nagappan, N. (2007, September). Usage and perceptions of agile software development in an industrial context: An exploratory study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (pp. 255-264). IEEE.

Boehm, B. W. (1984). Software engineering economics. *IEEE transactions on Software Engineering*, (1), 4-21.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61-72.

Bogdan, C., Gulliksen, J., Sauer, S., Forbrig, P., Winckler, M., Johnson, C., ... & Kis, F. (2016). *Human-Centered and Error-Resilient Systems Development* (Vol. 9856). Springer International Publishing.

Børte, K., & Nerland, M. (2010). Software effort estimation as collective accomplishment: An analysis of estimation practice in a multi-specialist team. *Scandinavian Journal of Information Systems*, *22*(2), 4.

Chaudhuri, D. J., & Chaudhuri, A. (2011). Agile burndown chart deviation-predictive analysis to improve iteration planning. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Cohen, L., Manion, L., & Morrison, K. (2017). *Research methods in education*. Oxon: Routledge.

Cohn, M. (2005). *Agile estimating and planning.* Pearson Education.

Cortina, J. (1993). What is coefficient alpha? An examination of theory and applications. *Journal Of Applied Psychology*, *78*(1), 98-104.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). A lightweight guide to the theory and practice of scrum. *Ver*, *2*, 2012.

Dora, S. K., & Dubey, P. (2013). Software development life cycle (SDLC) analytical comparison and survey on traditional and agile methodology. *National Monthly Refereed Journal of Research in Science & Technology*, *2*(8), 22-30.

Drescher, G., & Garbers, Y. (2016). Shared leadership and commonality: A policy-capturing study. *The Leadership Quarterly*, *27*(2), 200-217.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9-10), 833-859.

Evans, J. R., & Mathur, A. (2005). The value of online surveys. *Internet Research*, *15*(2), 195-219.

Fehlmann, T., & Santillo, L. (2010). From story points to cosmic function points in agile software development–a six sigma perspective. *Metrikon-Software Metrik Kongress* (p. 24).

Ferguson, E. S. (1971). The Measurement of the" Man-Day". *Scientific American*, *225*(4), 96-103.

Fischer, G. W. (1981). When oracles fail—A comparison of four procedures for aggregating subjective probability forecasts. *Organizational Behavior and Human Performance*, *28*(1), 96-110.

Galorath, D. D., & Evans, M. W. (2006). *Software sizing, estimation, and risk management: when performance is measured performance improves.* CRC Press.

George, D., & Mallery, P. (2018). Reliability analysis. In *IBM SPSS Statistics 25 Step by Step* (pp. 249-260). Routledge.

Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*, *3*, 22-23.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, *34*(9), 120-127.

Hughes, R. T. (1996). Expert judgement as an estimating method. *Information and Software Technology*, *38*(2), 67-75.

Idri, A., Abnane, I., Abran, A. (2017). Evaluating Pred(p) and Standardized Accuracy Criteria In Software Development Effort Estimation. *Journal of Software: Evolution and Process*, *4*(30), e1925.

Jansson, A. S. (2007). Software maintenance and process improvement by CMMI. *UPTEC STS07037 November Examensarbete*, *20*.

Jayaswal, B. K., & Patton, P. C. (2006). *Design for trustworthy software: Tools, techniques, and methodology of developing robust software.* Pearson Education.

Jindal, T. (2016). Importance of Testing in SDLC. *International Journal of Engineering and Applied Computer Science (IJEACS)*, *1*(02), 54-56.

Jørgensen, M. (2004). A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, *70*(1-2), 37-60.

Jørgensen, M., & Grimstad, S. (2008). Avoiding irrelevant and misleading information when estimating development effort. *IEEE software*, *25*(3), 78-83.

Jorgensen, M., & Grimstad, S. (2012). Software development estimation biases: The role of interdependence. *IEEE Transactions on Software Engineering*, *38*(3), 677-693.

Jørgensen, M., & Gruschke, T. M. (2009). The impact of lessons-learned sessions on effort estimation and uncertainty assessments. *IEEE Transactions on Software Engineering*, *35*(3), 368-383.

Jørgensen, M., & Sjøberg, D. I. (2004). The impact of customer expectation on software development effort estimates. *International Journal of Project Management*, *22*(4), 317-325.

Kerzner, H. (2017). *Project management: a systems approach to planning, scheduling, and controlling.* John Wiley & Sons.

Kim, T. K. (2015). T test as a parametric statistic. *Korean Journal of Anesthesiology*, *68*(6), 540-546.

Kumar, P. S., Behera, H. S., Kumari, A., Nayak, J., & Naik, B. (2020). Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. *Computer Science Review*, *38*, 100288.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. *International Conference on Information and Network Technology, 37*(1) 162-167

Lekh, R. (2015). Exhaustive study of SDLC phases and their best practices to create CDP model for process improvement. *International Conference on Advances in Computer Engineering and Applications* (pp. 997-1003). IEEE.

Lester, A. (2006). *Project management, planning and control: managing engineering, construction, and manufacturing projects to PMI, APM and BSI standards.* Elsevier.

Lett, S. H. (1998). An earned value tracking system for self-directed software teams. *Proceedings of European SEPG*, *98*.

Lingard, R., & Berry, E. (2002, November). Teaching teamwork skills in software engineering based on an understanding of factors affecting group performance. In *32nd Annual frontiers in Education* (Vol. 3, pp. S3G-S3G). IEEE.

Mallidi, R. K., & Sharma, M. (2021). Study on agile story points estimation techniques and challenges. *International Journal of Computer Applications*, *174*(13), 9-14.

MacDonell, S. G., & Shepperd, M. J. (2004, September). Using prior-phase effort records for re-estimation during software projects. *International Workshop on Enterprise Networking and Computing in Healthcare Industry* (pp. 73-86). IEEE.

Moløkken-Østvold, K., & Jørgensen, M. (2004). Group processes in software effort estimation. *Empirical Software Engineering*, *9*(4), 315-334.

Molokken-Ostvold, K., & Jorgensen, M. (2005). A comparison of software project overruns-flexible versus sequential development models. *IEEE Transactions on Software Engineering*, *31*(9), 754-766.

Moløkken-Østvold, K., Haugen, N. C., & Benestad, H. C. (2008). Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software*, *81*(12), 2106-2117.

Niessink, F., & Van Vliet, H. (1997, October). Predicting maintenance effort with function points. *Proceedings International Conference on Software Maintenance* (pp. 32-39). IEEE.

Ooi, G., & Soh, C. (2003). Developing an activity-based costing approach for system development and implementation. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, *34*(3), 54-71.

Prayitno, E., Siregar, J., Dewi, Y. N., Bachri, C., Indriyani, L., & Ma'arif, S. (2020, November). Use Case Points (UCP) with 3 Point in Program Evaluation and Review Technique (PERT) to Estimate Effort Software. In *Journal of Physics: Conference Series* (Vol. 1641, No. 1, p. 012008). IOP Publishing.

Ragunath, P. K., Velmourougan, S., Davachelvan, P., Kayalvizhi, S., & Ravimohan, R. (2010). Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). *International Journal of Computer Science and Network Security*, *10*(1), 112-119.

Rawsthorne, D. (2008). Monitoring scrum projects with AgileEVM and earned business value (EBV) metrics. *Agile Journal*, 33-45.

Rummel, R. J. (1988). *Applied factor analysis*. Northwestern University Press.

Ruxton, G. D. (2006). The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test. *Behavioral Ecology*, *17*(4), 688-690.

Saleh, S. M., Huq, S. M., & Rahman, M. A. (2019, February). Comparative study within Scrum, Kanban, XP focused on their practices. *International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-6).

Schwaber, K., & Beedle, M. (2001). *Agile software development with Scrum.* Prentice Hall PTR.

Sharma, S., & Pandey, S. K. (2015, August). Integrating AI Techniques In SDLC: Design Phase Perspective. *In Proceedings of the Third International Symposium on Women in Computing and Informatics* (pp. 383-387).

Srivastava, S., Trehan, K., Wagle, D., & Wang, J. (2020). Developer Velocity: How software excellence fuels business performance.

Tanveer, B., Guzmán, L., & Engel, U. M. (2017). Effort estimation in agile software development: Case study and improvement framework. *Journal of Software: Evolution and Process*, *29*(11), e1862.

Thomke, S., & Reinertsen, D. (1998). Agile product development: Managing development flexibility in uncertain environments. *California Management Review*, *41*(1), 8-30.

Turner, J. R., Turner, J. R., & Turner, T. (1999). The handbook of project-based management: improving the processes for achieving strategic objectives.

Turner, J. R. (2009). *Handbook of project-based management: Leading strategic change in organizations.* McGraw-Hill Education.

Tuteja, M., & Dubey, G. (2012). A research study on importance of testing and quality assurance in software development life cycle (SDLC) models. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(3), 251-257.

Usman, M., Mendes, E., Weidt, F., & Britto, R. (2014, September). Effort estimation in agile software development: a systematic literature review. *In Proceedings of the 10th International Conference on Predictive Models in Software Engineering* (pp. 82-91).

Živadinović, J., Medić, Z., Maksimović, D., Damnjanović, A., & Vujčić, S. (2011). Methods of effort estimation in software engineering. In *Proc. Int. Symposium Engineering Management and Competitiveness (EMC)* (pp. 417-422).

Zahraoui, H., & Idrissi, M. A. J. (2015, October). Adjusting story points calculation in scrum effort & time estimation. In *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)* (pp. 1-8).