Utrecht University

INVEST in Success:
A Deep Dive into User Story Optimization

Author
Ashot A. Grigorian

Student ID at Utrecht University:
6501435

Supervisor
Dr. Gerard Wagenaar

Second Supervisor
Prof. Dr. Fabiano Dalpiaz

A Master's thesis in Business Informatics
submitted to fulfill the requirements for
the Graduate School of Natural Sciences,
Department of Information and Computing Sciences

June 2024

Acknowledgments

First and foremost, I want to thank my supervisor, Dr. **Gerard Wagenaar**, for his support from the start to finish of this Master's thesis. He was always there when needed, providing exceptional guidance and encouragement that helped me through this challenging journey.

I also want to thank my second supervisor, Prof. Dr. **Fabiano Dalpiaz**, who provided valuable feedback throughout the study. Although feedback was not frequently requested, his expert insights were crucial in shaping this research whenever they were given.

Special thanks are due to **Antonie de Waele**, **Laura Fidalgo**, **Douwe de Haan**, **Bob van Heijster**, **Erik Hagen**, and **Nick van Ramshorst** for their support in onboarding projects. Your assistance was crucial during the recruitment process, for which I am grateful.

I am also thankful to **Sander van Nifterik**, **Stefan van den Eijkel**, and **Gil Lopes** for their regular check-ins and unwavering support. Your concern and encouragement were key motivators during this research process.

I want to express my gratitude to **Valentina Sargisian** for providing me with mental support over the final few months. The interest shown, and the encouragement provided were invaluable and helped me stay on track to complete this research.

Lastly, I extend my heartfelt thanks to my parents and siblings for showing unlimited support. Also for the belief in me, which turned into a constant source of strength throughout my academic journey.

To all who supported me, whether mentioned here or not, thank you.

Your support and encouragement made the completion of this thesis possible.

Contents

Abstract

Aiming to enhance Agile software development, this study evaluates the impact of the INVEST framework on the formulation of user stories and the impact on Agile software development project teams. We tackle issues like the constant adaptation to changing user preferences, which are common in fast-paced digital environments. It includes a literature review, where we explore user stories and discuss their structure, challenges, and importance. We evaluate how the INVEST framework's criteria play a role in writing user stories so they are clear, concise, and practical for development and testing. The study also reviews diverse methods to assess the quality of user stories, focusing on both quantitative and qualitative methods. Moreover, the research outlines a procedure to evaluate the practicality and impact of INVEST in real-world Agile software development projects. Through multiple case studies, the study found that applying the INVEST framework improved the clarity, value, and testability of user stories, though challenges remained in ensuring independence and negotiability. The empirical data suggest that teams using INVEST-aligned user stories experienced enhanced collaboration and more valuable refinement sessions. Besides these promising results, the study indicates that while INVEST is beneficial in improving the quality of user stories, its implementation requires substantial effort and adaptation to specific project contexts.

Keywords. INVEST Framework, User Story, Agile, Software Development, Requirements.

1. Introduction

1.1 The Role of Agile Methodologies in Modern Software Development

In today's fast-paced digital world, where not only society but also technology rapidly evolves and consumer demands shift almost overnight, traditional models of software development often fall short (Aitken & Ilango, 2013; Neumann et al., 2021). This presents us both a significant social and economic dilemma: How can businesses keep up with the accelerating pace of technological change and increasingly complex user demands? Traditional software development methods are not flexible enough to adapt swiftly to aggressive customer requirement changes, highlighting their shortcomings in this fast-evolving environment (Papadopoulos, 2015). One potential solution is the adoption of Agile methodologies, a transformative approach in software development that has reshaped how businesses respond to these challenges (Al-Saqqa et al., 2020; Altameem, 2015). Agile methodologies are not just about developing software; they represent a paradigm shift in thinking, emphasizing adaptability, customer-centricity, and rapid response to change – crucial elements in today's digital economy (Aitken & Ilango, 2013; Baham, 2016; Dingsøyr et al., 2012). This shift has given rise to a variety of specific methodologies, each in place to meet the challenges and demands of modern software projects.

Agile methodology is a collective term for various iterative and incremental software development methodologies, including frameworks like Scrum, Kanban, Lean, and Extreme Programming (XP) (Anderson, 2010; Schwaber & Sutherland, 2017). These methodologies exemplify the core principles of the Agile Manifesto, prioritizing individuals and interactions, working software, customer collaboration, and responding to change (Beck et al., 2001; Govil & Singh, 2022). Agile's flexibility and focus on continuous improvement make it suited to navigating the unpredictable waters of modern software development, including changing customer demands (Highsmith, 2009; Larman & Vodde, 2008).

Central to realizing these principles in practical terms are user stories, which serve as concise, simple descriptions of a feature from the user or customer's perspective, essential in translating customer needs into actionable development tasks (Ananjeva et al., 2020a; Cohn, 2004). They are crucial in ensuring the software development team understands the end user's needs and the context of the required functionality, thereby encouraging a user-centered approach to Agile Software Development (ASD) and enhancing team collaboration and adaptability (Lucassen et al., 2015; Patton, 2014; Rubin, 2012; Wake, 2003). Furthermore, user stories facilitate effective communication between clients and developers, and aid in creating a product backlog - a prioritized list of work derived from the project roadmap and its requirements, ensuring that the final product aligns closely with user needs and values (Schwaber & Sutherland, 2017; Rubin, 2012; Jeffries et al., 2000; Sutherland & Schwaber, 2013).

Still, crafting user stories in ASD comes with a challenge. When creating these user stories, one must ensure they are clear and concise, yet sufficiently detailed to avoid misaligned development efforts and to maintain project flexibility. Borhan et al. (2022) and Granda et al. (2021) have made this balance a central focus by emphasizing integrating user stories in ASD. These studies highlight the role of user stories in producing high-quality software within budget and time constraints (Borhan et al., 2022), and often focus on theoretical aspects or automated quality checks of user stories, potentially neglecting the human-centric aspects of their creation and implementation (Granda et al., 2021). This oversight suggests a need for more comprehensive research that considers the human elements in Agile processes.

Exploring this problem is particularly compelling because it lies at the intersection of several disciplines such as technology, business, and communication. The process of creating effective user stories involves a deep understanding of user behavior and expectations, coupled with the ability to translate these into viable technical requirements. It can be compared to a complex puzzle that requires insights into natural language communication, technical expertise, and business strategy. Addressing this challenge is not just about improving a technical process; it is about bridging the gap between human needs and technological solutions, leading to products that are both functionally robust and meaningful to users.

1.2 User Stories in Agile Software Development

User stories are a prominent technique for capturing and communicating customer requirements concisely and easily. They typically follow a simple format, called the Connextra template (Cohn, 2004; Lucassen et al., 2015), often articulated as: "As a [type of user], I want [an action] so that [a benefit/value is achieved]." Similarly, it can be formulated as "As a [persona], I [want to], [so that]." As can be noticed, with the second format, the third component is considered optional, which depends on the depth the user story attempts to achieve. This flexibility in formatting ensures that the story is focused on the user's needs and the desired outcome of the feature or functionality (Ananjeva et al., 2020b; Rehkopf, n.d.).

Crafting user stories in ASD requires a balance between clarity and detail. According to Borhan et al. (2022), it is crucial to integrate both non-functional and functional user stories to achieve high-quality software within budget and time constraints. Non-functional user stories (NFUs) detail the qualitative aspects of software, such as reliability, changeability, effectiveness, or accessibility, dictating how well the system functions. Functional user stories (FUs), on the other hand, describe the software's operational behaviors. This integration suggests that user stories must be comprehensive, covering various aspects of the software requirements, to ensure a thorough understanding of both the FU and NFU needs of the project (Borhan et al., 2022).

Writing effective user stories in ASD can be challenging. Lucassen et al. (2015) note that around 50% of real-world user stories have preventable linguistic defects, highlighting common issues in their formulation. This emphasizes the need for careful crafting of user stories to ensure they are clear, concise, and accurately reflect user needs. The purpose of various tools and methodologies in this field is to improve user story quality, focusing on enhancing the clarity and effectiveness of language to better meet user requirements.

A clarifying example from Dalpiaz and Brinkkemper (2021) to show the user story format is as follows: "**As a** conference attendee, **I want to** filter the talks by topic, **so that** I can attend those talks I am interested in". A less effective version of this user story might be: "**As** someone at the conference, **I** need some kind of tool to see different talks, **because** I want to see what I like." This version is problematic because it uses vague terms like "someone" and "some kind of tool", which do not clearly define the user role or the specific functionality needed. A developer, or individuals outside of the development team, might have trouble understanding this user story. Next to that, it lacks specificity about the desired action, merely stating "to see different talks" without explaining how this should be achieved functionally or what the exact requirement is, e.g. filtering by topic. This lack of clarity can lead to misunderstandings and misaligned expectations. While we do not want to prescribe the technical solution, we should still provide clear functional requirements to guide the development team. Lastly, it does not clearly articulate the benefit or the reason why the functionality is needed, merely mentioning "to see what I like" which is ambiguous and does not provide concrete value. While it might be optional within the Connextra template, including the benefit or reason is crucial for explaining the value of a user story to its reader.

Natural Language Processing (NLP) techniques have been identified as helpful tools in managing user stories, suggesting that they can aid in addressing some of the challenges in writing and organizing user stories (Raharjana et al., 2021). Additionally, the process of collaboratively writing user stories is influenced by factors such as the team's overall productivity and their experience in handling software requirements. This suggests that the way a team functions and its collective expertise can significantly affect the quality of the user stories they produce (Noel et al., 2018), meaning that a team with junior-level of experience might struggle more often than a team with senior-level of experience.

In summary, user stories are a crucial element in ASD, providing a framework for understanding and communicating customer requirements. The general structure and importance of clarity and detail in user stories are well-established, but they also present challenges that require thoughtful consideration.

1.3 Understanding the INVEST Principles

The INVEST framework, standing for Independent, Negotiable, Valuable, Estimable, Small, and Testable, is a guideline for optimizing user story quality (Buglione & Abran, 2013). Like others, this framework aims to create user stories that are well-defined and actionable, enhancing the efficiency of Agile projects. Recent research, such as that for Scrumlity, a modification of Scrum that focuses on enhancing ASD (Tona et al., 2022), supports the importance of INVEST in maintaining high-quality user stories. Within their study, they demonstrate the ongoing relevance of INVEST in Agile practices. Even so, despite its widespread acknowledgment in the Agile community and academic research, there is a noticeable gap in empirical evidence to assess and improve the quality of such frameworks (Lucassen et al., 2015).

Before tackling specific sub-research questions, we need a better understanding of INVEST in the context of user stories. For this, we explore the framework and its acronym further. Along with that, we need to grasp how to measure whether the elements are adhered to within user stories. To achieve that, we search for methods and techniques with which each element can be analyzed separately. In this pursuit, we turned to insights from literature. Within this context, the study by Cowperthwaite et al. (2023) notes that if a user story does not meet one or more of these criteria, it is not considered of good quality.

1.3.1 **I**NVEST: **Independent**

User stories should be independent, meaning each one must be analyzed individually to determine if it is sufficiently self-contained to be independent of others (Ferreira et al., 2022; Halme et al., 2021). This ensures a user story can be moved to another Sprint without impacting software deployment. Buglione and Abran (2013) mention that establishing a threshold or tolerance level for all criteria can help reduce subjectivity in its evaluation. For the Independent criterion, they mention a user story is considered sufficiently independent if it does not need to be split into more sub-requirements.

Martakis and Daneva (2013) recognize in their study that while the INVEST criteria advocate for independence in user stories, dependencies can be inevitable in complex projects. Those claims were substantiated by their focus group experiment. The participants, who were experienced in handling requirements in ASD, provided insights into real-world practices and challenges. They substantiated the idea that dependencies, despite the preference for independence in user stories as per the INVEST criteria, are often unavoidable in complex projects. They suggest that effective management of these dependencies is essential, focusing on regular communication and a hybrid Agile-plan-driven approach to mitigate risks (Martakis & Daneva, 2013). This shows how Agile principles are adapted for large-scale software projects, understanding that in practical situations, complete independence might not be possible because user stories are interconnected and complicated.

1.3.2 I**N**VEST: **Negotiable**

For user stories to be negotiable there should be flexibility for the development team to negotiate changes before they become full requirements (Buglione & Abran, 2013). The provider, which is often considered to be the Product Owner (PO), needs to allow changes while avoiding scope creep or time constraints related to the agreed schedule for modifying user stories. These negotiable elements could be changes in the scope, approach, or priority. Having negotiable user stories involves managing changes in a way that does not lead to an uncontrolled increase in project scope, which can happen if too many modifications are made without proper review or alignment with the project's objectives and constraints (Buglione & Abran, 2013). Put differently, user stories are not explicit contracts: They should remain open for discussion and adaptation as the project progresses and more information becomes available (Ferreira et al., 2022). To keep this feasible, the user story should not be overly detailed (Halme et al., 2021).

1.3.3 IN**V**EST: **Valuable**

Every user story should deliver clear and tangible value to end-users (Ferreira et al., 2022), which could be purchasers or users (Halme et al., 2021). This ensures that development efforts are aligned with user

needs and contribute to the overall objectives of the project, to make user stories valuable to the client. The value they hold to the client is often retrieved from the optional component within user stories.

1.3.4 INV**E**ST**: Estimable**

To be estimable means a user story should be clear enough that developers can reasonably estimate how much effort, time, and resources it will take to complete the user story (Ferreira et al., 2022; Halme et al., 2021). This is crucial for effective planning and resource allocation.

1.3.5 INVE**S**T: **Small**

The Small element of the INVEST criteria is about the size and scope of a user story in ASD (Buglione & Abran, 2013). Each user story should be just the right size - not too big or too small. This criterion ensures they are sufficiently granular but not overly complex or too high-level. This means that user stories must be manageable and can be completed within a Sprint, without being so small that it becomes inefficient or trivial to handle (Buglione & Abran, 2013). Having this balance ensures that they are practical to work on and contribute meaningfully to the project's progress within each Sprint, both regularly and incrementally (Ferreira et al., 2022). It is also mentioned that smaller user stories are more easily estimated, further emphasizing the importance of this criterion (Halme et al., 2021). User story splitting is a suggested technique for breaking down larger user stories into smaller, more manageable parts. This approach aligns with INVEST, particularly in making user stories concise and manageable, thereby enhancing their quality and effectiveness in Agile project management (Dellsén et al., 2022). Two main techniques for splitting user stories are horizontal splitting, which divides user stories by architectural layers like UI, backend, and storage, and vertical splitting, which focuses on a single function across these layers.

Ernst et al. (2015) suggest a methodical approach to breaking down Quality Attribute Requirements (QARs) in large-scale ASD. QARs are defined as specific, non-functional requirements that software must meet to ensure its quality and often include aspects like security, reliability, performance, maintainability, and usability. They recommend dividing these high-level requirements into smaller, more tangible parts, making them easier to integrate into iterative development cycles. This process involves detailed analysis and planning to ensure that each component of the QARs is actionable and fits within the Agile framework, facilitating a balance between comprehensive quality management and Agile flexibility (Ernst et al., 2015).

1.3.6 INVES**T**: **Testable**

The Testable criterion underlines the need to incorporate elements that allow for testing the user story. This is important to reduce the Cost of Non-Quality (CONQ) beforehand, instead of focusing only on more visible costs that may be incurred before the first deployment. It implies that user stories should be formulated to allow for effective testing and validation of features (Buglione & Abran, 2013). This can be realized by attaching acceptance criteria (Ferreira et al., 2022). These are clear criteria to test whether the user story has been successfully implemented. This ensures that the user story's objectives are met and that the implemented feature works as accurately as intended (Halme et al., 2021).

1.3.7 The INVEST Grid

In the study conducted by Buglione and Abran (2013), the INVEST grid is proposed as a structured approach to evaluate and improve user stories. When discussing the INVEST grid in the study of Khanh et al. (2017), it is mentioned the grid supports forming a holistic view of requirements. The grid, depicted in Figure 1, serves as a practical template for practitioners to assess the quality of user stories before they are considered ready for implementation in a Sprint (Dellsén et al., 2022). Within this grid, the acronym INVEST is cared for, with each element representing a fundamental characteristic of a well-formulated user story. The grid rates user stories by assessing each element of INVEST based on a four-point scale. The lowest score, 0, indicates that the attribute is poorly represented or missing. On the other end, the

highest score, 3, suggests that the attribute is presented exceptionally. Scores of 1 and 2 fall between these extremes, showing incremental improvements from poor to excellent.

INVEST	Description	0	1	2	3
		Poor /Absent	**Fair**	***Good***	***Excellent***
I – Independent	*User Stories should be as independent as possible*	The start of construction of a US is tied to the completion of at least one other US	The completion of a US hinders the start of construction of at least one other US	The US can contain any constraint, but its release can be constrained by the completion of at least one other US	The US is fully independent, and it can be realized and released with any constraint
N – Negotiable	*User Stories should be "open", reporting any relevant details as much as possible*	The US contains enough detail to be a technical specification (Design phase), leaving no room to negotiate any element	The US is written with enough detail to be a functional specification (Analysis phase), leaving no room to negotiate any element	The US is written with informative content defining a User Requirement in a consolidated manner, yet shared between Customer and Provider	The US is written with the informative content typical of a high-level need, allowing feedback between customer and provider
V – Valuable	*User Stories should provide value to end users in terms of the solution*	The functional part (F) of the US does not contain all the functionalities requested by the customer	The functional (F) part of the US expresses mostly qualitative (Q) and technical (T) requirements about the system, and needs to be more developed in terms of functional requirements	The functional (F) part of the US expresses mostly the functional requirements requested by the Customer, but also includes qualitative (Q) and technical (T) requirements	The functional (F) part of the US correctly expresses only the functional requirements requested by the customer
E – Estimable	*Each User Story must be able to be estimated in terms of relative size and effort*	The US shows only its functional (F) part, filled in by the customer, but without sufficient detail to allow the provider to fill in the Q/T parts	The US shows only its functional (F) part, filled in by the customer, but validated with the provider	The US has been completed by the provider with respect to Q/T issues, but still needs to be validated jointly with the customer	All the useful parts of the US (F/Q/T) are shown, allowing the effort need to size and estimate it, and validated by both parts
S – Small	*Each User Story should be sufficiently granular, and not defined at too high a level*	The US is very large, and cannot be completed within a Sprint	The US is very large, and can be completed within a Sprint, but cannot accommodate the creation/delivery of other US	The size of the US is such that it can be completed within a Sprint, jointly with other US, but it is too small to create overhead about the Testing phase	The size of the US is such that it can be completed within a Sprint, jointly with other US, ensuring an appropriate balance between development and testing activities
T – Testable	*Each User Story must be formulated in an effort to stress useful details for creating tests*	The US does not include tips about Acceptance Tests	The US includes a formal indication of Acceptance Tests, but yet to be completed	The US includes an indication of Acceptance Tests which are complete, but yet to be validated	The US includes an indication of completed and validated Acceptance Tests

Figure 1: The INVEST Grid (Buglione & Abran, 2013)

Besides supporting the preliminary stages of Sprint planning, this assessment grid also promotes continuous improvement. By systematically addressing each criterion, Agile teams can iteratively refine their user stories, thus enhancing the quality and effectiveness of their software development processes. The INVEST grid, therefore, is not just a checklist but a dynamic tool for growth and excellence in ASD.

Taking into consideration the example user story from Dalpiaz and Brinkkemper (2021) once more, in combination with the INVEST grid, a reworked version of this user story to better adhere to INVEST would be: "**As a** conference attendee, **I want** a feature to filter the scheduled talks by their topic through the conference app, **so that** I can efficiently plan my attendance at sessions that align with my interests." Figure 2 illustrates this mockup user story, where all INVEST criteria are considered.

User Story #1234 - Scheduled Talks Topic Filtering

As a conference attendee, **I want** a feature to filter the scheduled talks by their topic through the conference app, **so that** I can efficiently plan my attendance at sessions that align with my interests.

Functional Description

The proposed feature in the conference app allows attendees to filter talks by selecting specific topics, helping them to easily find and plan their attendance at sessions of interest. It includes an option to reset the filters, allowing users to switch back to viewing all talks. This user-friendly feature is designed for efficiency and ease of navigation within the conference schedule.

Acceptance Tests

1) When the user selects topics from the list in the conference app, the app should display only the talks associated with those selected topics. For instance, if a user selects "Artificial Intelligence", the app should show only the talks related to Artificial Intelligence.

2) The user should have the option to reset the selected filters. After resetting, the app should display all talks irrespective of topic.

Figure 2: A Mock-Up User Story Adhering to INVEST

The reworked user story of Figure 2 is designed to be independent. This means that the development team can work on it without having to wait for other parts of the project to be completed, making the process more efficient. We see this from the lack of dependencies to other user stories mentioned in the example. Although dependencies could potentially have been left out of this user story while existing in practice, in the current state it would receive a 3/3 score for Independent. The user story is also open to interpretation, as it is not described in detail how the filter should be created, rather high-level, allowing flexibility and room for negotiation. This means the team has the creative freedom to come up with the best solution during the development phase. From this, it is considered negotiable with a 3/3 score.

Importantly, the feature described in Figure 2 is something that the users will find valuable. It is directly tied to their goal of making the most of their time at the conference by focusing on talks that interest them, which is exactly what these users are looking for, showing why it would receive a 3/3 score for Valuable. In terms of estimating the effort required, the team can draw on past experiences with similar tasks to predict how much work will be involved in creating the filtering feature. Besides that, sufficient detail is presented from a functional perspective for the developers to come up with a technical solution, without ambiguities present. This makes planning and allocation of resources straightforward, allowing a 3/3 score.

Then, the scope of the user story is kept intentionally narrow to ensure it is manageable and can be completed on time, which is particularly important if the development team is working within a tight timeframe, such as that of a Sprint. By keeping the score narrow, and limited to a single functionality, we ensure the user story is well-sized to fit within a single sprint, also allowing sufficient time for both testing activities and other user stories. Following this, the Small criterion would score 3/3 for this example user story of Figure 2.

Lastly, the functionality's success can be measured through testing. The team can create specific tests to confirm that the filter works as intended, showing users only the talks that match their interests. Examples of those tests are presented in Figure 2 under the header "Acceptance Tests" where not only the intended behavior is explained, but also examples are given to ensure the entire team is thinking in the same direction when picking up this user story. By considering both the happy and an alternative flow, this user story seems to completely cover the different flows, or scenarios. By considering the end-user, who would be a conference attendee, would want these scenarios to be taken care of during testing, we can assign a 3/3 score for Testable as well.

2. Research Methods

2.1 Research Procedure

We explore the use of the INVEST framework in real-world projects to address the complexities of formulating effective user stories in ASD. It explores strategies to optimize user story quality according to the INVEST criteria, recognizing the critical role of user stories in aligning developments with user needs and project objectives. The primary focus is on understanding the balance between technical requirements and human-centric aspects in the creation of user stories, and how frameworks like INVEST influence this process. By examining industry-specific cases, the study offers insights into the value of applying INVEST to improve user story practices, thereby enhancing the overall quality and success of Agile project efforts.

Our study will combine literature reviews, interviews with Agile practitioners, and experiments to assess the INVEST framework's real-world effectiveness, as visualized in Figure 3.

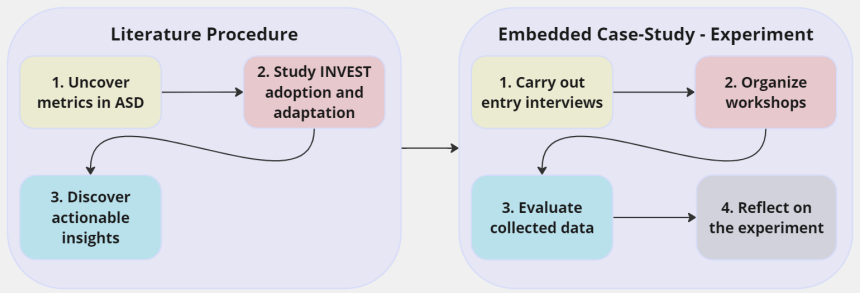

Figure 3: The Research Procedure

Initially, we review the literature for theoretical insights, followed by interviews for current industry perspectives. This literature review will focus on case studies that have considered the INVEST framework to uncover potential pitfalls in experimenting with the framework. Along with that, techniques will be considered to both study the INVEST framework in practice and to draw comparisons between case studies. The empirical phase of this research, where we study Agile projects, will analyze how INVEST is applied to existing user stories. Then, we conduct various expert interviews to understand the current state of their projects. We follow up with workshops, with these Agile professionals, to adjust user stories to strictly follow INVEST, creating control and experimental groups for comparison. This method aims to provide practical, empirical data on INVEST's impact on Agile software project development.

The primary objective here will be to monitor the anticipated versus actual completion process of these user stories, compared to those in the control group. To follow up on that, the main research question is formulated in such a manner where both the results gathered from these pre-existing literature studies and the experiments of this study are relevant. The following main research question is investigated:

> **Main Research Question:** *"How does the INVEST framework impact the practical usefulness of user stories in real-world Agile software development projects?"*

In this context, we refer to "practical usefulness" as the tangible benefits and improvements in user story quality, team communication, and project adaptability that result from applying the INVEST framework in real-world Agile projects.

Finally, we draw actionable insights from the analysis of case studies where the INVEST framework has been applied. This approach is particularly relevant as it allows us to extract practical lessons and strategies from real-world examples. By studying these case studies, we gain a deeper appreciation of

how the INVEST criteria function in practice, offering valuable lessons that can be applied to enhance the performance of user stories in ASD.

As such, this research holds a great scientific contribution by generating new empirical data that can support, challenge, or refine existing theories and hypotheses about the role of user stories and the impact of frameworks like INVEST on such methods in practice. This expansion of academic understanding is crucial for further enhancing Agile practices and the effective integration of user stories into ASD. According to Yin (2009), the case study method is particularly suited for capturing the complexity of real-life contexts, which is critical for understanding how theoretical frameworks like INVEST function in practice. Similarly, Eisenhardt and Graebner (2007) emphasize the importance of building theory from case study research, as it allows for the development of deeper insights and more robust theories.

Empirically, the study aims to enhance the efficiency, productivity, and sustainability of ASD projects. By offering actionable strategies and best practices for user story utilization, it promises to impact how we plan and execute projects, leading to more successful, user-aligned software. The use of mixed-method approaches, combining qualitative and quantitative data as suggested by Creswell and Plano Clark (2017), ensures reliable findings for both academic and practical purposes in ASD.

2.2 Literature Topics

In the following chapter, we focus on a structured procedure to study the existing scientific literature on the INVEST framework. To address the overarching main research question, “How does the INVEST framework impact the practical usefulness of user stories in real-world Agile software development projects?” The review focuses on three specific sub-research questions, each providing a different perspective through which the main question is explored.

The exploration of literature begins with a thorough analysis of different approaches to consider when measuring the success and efficiency of the INVEST framework. This involves an analysis of both quantifiable data and subjective evaluations, offering a comprehensive view of how to assess its impact in practical scenarios. Simultaneously, we illustrate a range of real-world examples from literature to understand these principles in action, providing a tangible context for their application in dynamic Agile settings. This analysis aims to provide us with actionable insights and a deeper appreciation of how the INVEST framework functions in the real world, offering valuable lessons and takeaways for the experiment of the current research.

The first sub-research question (SRQ 1) of this study, “What are effective methods to measure the impact of user stories that adhere to the INVEST criteria on Agile teams?”, is crucial in establishing a direct link between the INVEST framework and the impact of user stories that adhere to the framework. Exploring and identifying effective measurement methods allows us to quantify the impact of INVEST on user stories, thereby directly addressing the core of the main research question. Along with that, it allows us to build on that knowledge by applying the best practices in this research experiment.

The second sub-research question (SRQ 2), “How are the INVEST principles adopted and adapted in diverse Agile software development settings?”, shifts the focus to the practical application of the INVEST principles. To answer this question, we discuss case studies where INVEST has been applied previously, providing insights into its adaptability and effectiveness across various contexts. Understanding these variations and their impacts is essential for assessing how the INVEST framework influences the overall performance of user stories in a range of Agile environments.

Lastly, the third sub-research question discussed in the literature review (SRQ 3), “What actionable insights can be drawn from the analysis of the INVEST framework’s application in case studies?”, involves extracting practical lessons and strategies from the discussed real-world examples. While SRQ 2 provides a foundation by illustrating the varied applications and adaptability of INVEST, SRQ 3 builds on

this by extracting concrete recommendations and strategies based on these applications. This helps us see how INVEST can be used not only in theory but also in real project settings to improve teamwork and project management.

2.3 Literature Collection

In conducting the literature review, we are deliberately choosing an open timeframe. This means we will consider published works from any period, without imposing any specific time frame-related limitations. Yet, random samples indicate that the majority of literature originates from the period beyond 2010. This approach is designed to ensure a thorough understanding of the topics at hand, by including a broad perspective on the current state of research on INVEST. By not restricting the review to certain key events, we avoid the exclusion of any, perhaps, highly relevant studies. While not imposing time frame restrictions, there are language restrictions in place for the sources considered. The review will be restricted to sources written in English. This approach is in line with the dominant language used in scientific studies within the fields of computer science and project management. It guarantees that the sources we review are both accessible and relevant to the research questions.

Quality assessment criteria for the studies will include the publication in reputable journals or conferences, the expertise of the authors, and the rigor of the methodologies employed. The impact of a study will not be considered, that is its citation count, considering there is a lack of empirical studies on this topic. To not be limited to the coverage of specific databases, the literature search will be conducted using the Google Scholar search engine, known for its broad coverage of academic literature. This engine is chosen for its extensive repositories of technical and scientific literature, especially in the areas of computer science and project management.

The literature search will be carried out using keywords directly related to the research topic. For this, we consider keywords such as "Agile", "INVEST framework", and "user story". These terms are iteratively refined and adjusted based on the relevancy and results of initial searches, ensuring a comprehensive coverage of the topic. To effectively execute this, a foundational search query was established that could be adapted to explore the various sub topics mentioned previously. The process involved an iterative refinement of various search terms, aiming to create a base query that is both specific enough to retrieve relevant results and broad enough to accommodate additional subtopic-related terms.

Initially, we considered the following query: `"invest"`. The purpose was to gain a broad understanding of the term's usage and relevance across various contexts. However, the initial search was overly broad, mostly retrieving results unrelated to Agile methodologies, with around 6,110,000 results. This result was expected due to the common use of "invest" as a verb in English. This made us narrow the focus to `"invest framework"`. While this significantly reduced the number of irrelevant results, the content was still heavily skewed towards financial aspects, with 82 results. Then, we adjusted the query to `"invest framework" AND "agile"` and also `"invest framework" AND "user stories"` which gathered 21 and 17 results, respectively. While this query is more specific, the number of results were limited, indicating the need for a broader yet relevant approach, and also showcasing the current scientific gap on this topic. This resulted in the final base query: **`("invest criteria" OR "invest framework") AND ("agile" OR "user stories")`**, with 148 search results. This query serves as a foundation for the systematic exploration of subtopics. Of these 148 results, 145 results stem from 2010 - 2023.

For `SRQ 1`, we change the base query to focus on **("invest criteria" OR "invest framework") AND ("agile" OR "user stories") AND ("measurement" OR "evaluation" OR "metrics") AND "performance" AND "assessment"** - defined as Query 1, with 49 search results. This helps us find articles and studies that talk about how to measure and evaluate the success of user stories in the Agile setting using the INVEST framework.

For `SRQ 2` we modify the query to **("invest criteria" OR "invest framework") AND ("agile" OR "user stories") AND ("case study" OR "case studies") AND "project"** - Query 2. This new search, with 83 results, helps us find information about how the INVEST framework is being used in different Agile projects. We want to see the different ways it is put into action and how this affects the results.

For `SRQ 3` the query is **("invest criteria" OR "invest framework") AND ("agile" OR "user stories") AND ("lessons learned" OR "best practices" OR "strategy") AND "implementation"** - defined as Query 3. With the 81 results that are retrieved from this search query, we look for real-world examples and case studies where the INVEST framework has been used. This is about understanding the practical side of things and seeing what we can learn from actual projects that have used the INVEST framework.

By adjusting our search for each sub-question, we ensure a thorough literature review focused on understanding how the INVEST framework affects user story performance in Agile projects. This methodology closely adheres to systematic mapping study principles, including structured literature collection, classification, and analysis, though it lacks visual mapping and multiple database searches typically found in SMS.

3. Literature Review

3.1 Metrics of Success for User Story Development

Previously, we have examined both user stories and the various elements of the INVEST criteria. Our study now turns to exploring the essential criteria and benchmarks that determine the effectiveness of user stories in software project management. We discovered the need for quantifiable measures to assess the impact of user stories on project progression and outcomes. By analyzing various evaluation methods, we aim to provide insights into effective strategies for tracking the success of user stories.

This investigation is necessary to establish a tangible link between the aspects of the INVEST framework and the overall effectiveness of user stories. By focusing on measurement techniques, we aim to quantify the impact of the INVEST elements on the performance of user stories. This step is crucial in addressing the central question of our research and lays the groundwork for applying these insights in our experiment. To tackle `SRQ 1`, we employ but do not limit ourselves to Query 1. This approach is aimed at thoroughly exploring and identifying methods to quantify the impact of the INVEST criteria on the success of user stories in an Agile environment.

In a study by Willamy et al. (2016), burndown charts and velocity tracking are discussed for assessing product development, particularly from the perspective of the PO. Although not presenting an extensive description of burndown charts, the research involves analyzing the release burndown to describe the progress of developments. Such a chart is visualized in Figure 4 following more exploration of burndown charts and their application. Insight into the Sprint burndown helps teams predict their sprint's completion by visually comparing completed work to remaining work and highlighting scope changes after the sprint starts (Fuksmane, 2023).

Figure 4: A Sprint burndown presenting the remaining work for a two-week sprint and its suggested guideline (Fuksmane, 2023)

Willamy et al. (2016) highlight the use of burndown charts as a visual representation tool for mapping out the rate of work completed against the remaining workload. In their work, velocity tracking is associated with various metrics like estimation conformity, risks, and the number of open bugs. The study does not dive into the specifics of how velocity tracking is implemented, but it generally signifies the measurement of the team's work pace, often quantified in story points (SPs) over a sprint. The inclusion of velocity

alongside metrics concerning risks and bug counts implies an integrated approach to project assessment, highlighting its importance in a broader evaluative framework for ASD projects.

In a similar project, by Lucassen et al. (2017), the effectiveness of the Grimm Method was explored. While the Grimm method is not the primary focus of this study, the assessment techniques of their study are highly relevant. They came up with the strategy to combine exit surveys and interviews for qualitative insights into practitioners' experiences to understand the perceived impact on the work. Alongside this, they adopted specific metrics from Davis (2015) for a more objective analysis. These metrics, including communication frequency, rework levels, and defect rates, offer a comprehensive framework for evaluating methodologies in ASD (Lucassen et al., 2017). Defect rates, mentioned as the "Issue Count" of an iteration, appear to be a common theme in progress tracking, with the study of Kamath (2023) also underpinning this idea. While the study mentions the use of process metrics to objectively assess the impact of the Grimm Method, it does not go into detail about the data collection methods for these metrics.

The study by Adali (2017) investigated the correlation between Function Points (FPs) and SPs in ASD. FPs are a metric used to measure the size and complexity of software functionality, offering an objective way to quantify functionality regardless of implementation technology. Adali's research, using data from a Dutch banking organization, reveals that the relationship between FP and SP is inconsistent across different datasets. This finding cautions against generalizing their correlation, emphasizing the need for context-specific analysis in software metrics. However, it still offers a quantifiable technique to measure the complexity and required effort of a user story objectively. With SPs giving subjective estimations, FPs might prove to be more reliable.

Research mentions that in addition to FP and SP, other methods for analyzing user story progress in Agile development include Functional Size Measurement (FSM) (Adali, 2017). FSM is a standardized approach, with guidelines improved and monitored by organizations such as IFPUG and NESMA, and is recognized as an ISO standard: ISO/IEC 14143-1:2007 (International Organization for Standardization, 2007). FSM is introduced as a method that addresses the shortcomings of previous software sizing techniques. The technique shifts focus from how the software is built to measuring its size based on the functions that users need. This change in perspective allows for a more user-centric approach to determining the size of software (International Organization for Standardization, 2007). On a related note, Huijgens and Solingen (2014) highlight that Agile team velocities measured in SPs are subjective and vary significantly across different teams. This variation is due to SPs being based on each team's unique experiences and expertise. In contrast, FPs offer a more objective and standardized measure. As a result, SP cannot be reliably compared or standardized across teams, making them unsuitable for assessing productivity or performance on a broader organizational scale (Huijgens & Solingen, 2014). However, they can be effective within individual teams as a relative measure of effort and complexity. This approach allows teams to refine their estimation process based on their unique dynamics and historical performance. This finding emphasizes the importance of contextual understanding in applying Agile metrics like SP within specific team environments.

In this subchapter, we examined diverse methods to measure user story success in ASD, for which the studies are also required to be relevant to the INVEST framework. The range and depth of the references and their respective insights were retrieved using Query 1. With `SRQ 1` as follows "What are effective methods to measure the impact of user stories that adhere to the INVEST criteria on Agile teams?", it becomes evident that the covered studies appear to sufficiently and effectively address the gap of knowledge. The studies mention various quantitative and qualitative methods to measure user story impact in ASD. This includes practical tools like burndown charts, techniques such as exit surveys and interviews, tracking the issue count and complexity, and standard measures like Function Points, Story Points, and Functional Size Measurement. Together, they provide a comprehensive overview of

measuring techniques that align with the research question, addressing how one can measure user story impact on project teams.

3.2 Implementations of the INVEST Framework in Agile Contexts

Moving on from user story assessment metrics, we are now focusing on the application of INVEST in different Agile settings and observing the variations in its implementation. We will examine case studies to gain insights into how adaptable and effective the INVEST principles are across various contexts. Understanding how different applications of INVEST affect the quality and success of user stories in Agile environments is crucial. For this exploration, we deploy the search query labeled as Query 2. This search combines terms like "invest criteria" with "agile", and now also includes keywords such as "case study" along with "project" to find case-specific studies where INVEST is applied. With 83 results from this search, we are prepared to investigate the various ways the INVEST framework is applied in Agile projects and the impact of these different methods on the outcomes of the projects.

Continuing our analysis of literature, the formerly mentioned study of Dellsén et al. (2022) shows that while Agile teams understand and aim to follow the INVEST criteria, practical challenges lead to variations in application. The research features five case studies, with a different company for each study. These companies vary in size, domain, and geographic location, providing a diverse range of perspectives on Agile methodologies. The study focuses on how these companies implement the INVEST criteria in their processes, especially concerning user story splitting. These case studies offer valuable insights, discussing that teams often focus on creating small and testable user stories, which helps in better sprint planning and ensures effective quality assurance (Dellsén et al., 2022). However, the aspects of independence and negotiability in user stories are sometimes less emphasized, reflecting a compromise between ideal INVEST criteria and the demands of their fast-paced Agile project environments. This indicates a flexible application of INVEST, adapting to project-specific requirements and constraints.

Adding to this perspective, in the study by Lucassen et al. (2017), 30 practitioners from three different companies participated to explore the impact of various methods on user story quality. Afterward, by combining both quantitative and qualitative methods, they analyzed project metrics like velocity, bugs, and rework, alongside surveys and interviews to measure perceptions of story quality and team communication. They found improved intrinsic quality in user stories and more constructive team discussions, but no significant changes in project management metrics (Lucassen et al., 2017). From this, we understand that although certain methods seem to improve the quality of work and lead to more constructive user story conversations, reducing unnecessary rework, it does not always translate into improved efficiency.

Expanding on these findings, the study by Rizkiyah et al. (2020) emphasizes broader Agile methodologies and challenges in government outsourcing projects. This study focuses on the challenges and best practices in Agile project management, user story creation, and quality assurance. The challenges they discuss, such as adapting to changing requirements and ensuring good communication in Agile projects, highlight the importance of flexibility and clarity in Agile user stories. These points connect to INVEST, specifically emphasizing the necessity for user stories to be flexible (Negotiable) and clearly defined (Estimable). By emphasizing certain aspects, rather than all aspects, we see that there is a division between literature and practical applicability as not all aspects of INVEST are considered equally relevant within ASD in practice.

In a similar study by Kuhail and Lauesen (2022), the focus is on the practical challenges of applying quality criteria, similar to those of INVEST, in Agile settings. Specifically, these are applied in the context of user story quality. They analyzed user stories in a hotline system project, which is a system designed to handle a high volume of incoming phone calls, by assessing them against criteria like completeness, correctness, verifiability, and traceability. Their findings reveal that many user stories do not adequately capture complex project requirements, highlighting gaps in implementing the INVEST criteria Negotiable

and Estimable. This contributes to the understanding that while Small and Testable are often prioritized, the Independent and Negotiable criteria may not receive as much emphasis in real-world Agile environments, which in its turn complicates the Estimable criterion.

The study by Do Nascimento et al. (2022), which could be considered less related due to its more technical nature, yet highly relevant, used a mathematical model to evaluate user stories. This model was used to quantitatively assess the alignment of user stories with the INVEST criteria in ASD, providing a data-driven and objective approach to improve (Agile) project management practices. It involved a survey with a development team, analyzing four user stories by nine experts. By ensuring that user stories better meet the INVEST criteria, the study suggests that project management in Agile settings becomes more efficient and effective (Do Nascimento et al., 2022). This is inferred from the enhanced clarity and prioritization of user stories. Nonetheless, with only nine experts, it remains to be seen whether these results are generalizable for ASD as a whole.

Then, there was the study by Martakis and Daneva (2013) which primarily focused on handling requirements dependencies in Agile projects rather than directly on the INVEST criteria, as noticeable with the other studies discussed. However, the INVEST criteria emerge as a significant aspect in the context of managing these dependencies. The researchers conducted focus group discussions with Agile practitioners to gather insights. They conclude that while all INVEST criteria are crucial, the Independent and Testable criteria are more challenging to assess in the presence of dependencies (Martakis & Daneva, 2013). The study reveals that Agile teams often adapt INVEST criteria according to project-specific needs, emphasizing the importance of flexibility and continuous improvement in Agile methodologies, as also observed from other studies.

The literature review conducted in this subchapter, through Query 2, reveals that the INVEST criteria, analyzed through different Agile methodologies are applied with significant variation, adapting to the unique demands of each project. Agile teams often prioritize creating small and testable user stories, but underemphasize aspects like independence and negotiability, particularly when working in and with fast-paced environments. This makes clear the need for a more adaptable and flexible application of the INVEST framework, aligning it with the unique challenges and needs of various Agile projects. In addressing **SRQ 2**, defined as “How are the INVEST principles adopted and adapted in diverse Agile software development settings?”, these findings show the adaptability, effectiveness, and impact of INVEST across different contexts. The key insight is that variations in implementing INVEST principles, especially in balancing different aspects of user stories, critically influence the overall performance of these user stories in Agile settings.

3.3 Effectively Applying INVEST in Practice

In this chapter, we go beyond looking at how the INVEST framework is used, which we covered under **SRQ 2**. Instead, we focus on **SRQ 3**, aiming to find the best practices and important lessons learned from using INVEST in ASD. As with Query 2, we explore case studies and research papers. However, with Query 3, we aim to understand what are effective ways to apply INVEST when creating user stories. Within this subchapter, we address the challenges Agile teams face when adopting the INVEST framework. We focus on practical strategies and best practices derived from ASD case studies, aiming to improve team dynamics and project management in real-world settings.

Certain studies, such as the one by Pokharel and Vaidya (2020) primarily focus on the current state of understanding and usage of user stories in Agile methodologies, rather than providing specific strategies for more effectively applying the INVEST criteria in user stories. While they highlight the gap in effective application and comprehension of user stories, the emphasis is on the general need for improved training and education in Agile principles and user story writing. The study stresses the importance of adhering to the INVEST criteria but does not discuss strategies or considerations for enhancing the application of INVEST in user stories. Likewise, the study by Martakis and Daneva (2013) emphasizes

the importance of managing requirements dependencies in Agile project management, similar to traditional projects. It highlights the critical role of risk management, and individual responsibility, and particularly stresses the need for continuous communication and collaboration to mitigate risks. While the study acknowledges the challenges in adhering to the INVEST criteria for user stories, especially in making requirements estimable due to dependencies, it also points out the absence of systematic methods in Agile methodologies for handling these dependencies (Martakis & Daneva, 2013). This research is important in illustrating the complexities of applying the INVEST criteria in Agile settings and the necessity of an effective architecture approach for managing dependencies, without providing specific implementation guidelines.

Similarly, the study by Anitha et al. (2013), which focuses on managing requirement changes in projects that use Agile methods, highlights the importance of balancing technical approaches with team dynamics and culture. It emphasizes strategies like Agile training and empowering teams as key for effectively applying the INVEST framework in Agile environments. However, the study does not provide detailed guidance on how to conduct Agile training or specific ways to empower teams. Its main contribution lies in pointing out the necessity of these strategies in real-world Agile project management, rather than focusing on specific methods (Anitha et al., 2013).

Opposed to the INVEST criteria, results from the study by Jurisch et al. (2017) indicated that recommendations were most effective when they relied solely on the text of user stories. In comparison, including acceptance criteria in these recommendations actually led to a decrease in their precision. They demonstrated the value of user story text in Agile by assessing a mobile app development recommender, using information retrieval to assess effectiveness based on different elements of user stories. Jurisch et al. (2017) analyzed 84 user stories, created mostly by students, and added 60 more from an external dataset. However, the reliance on a student-created dataset and the limited focus on text similarity as the basis for recommendations could be seen as potential limitations of their research approach. The suggested best practice, of excluding acceptance criteria, does not fully align with the broader principles of INVEST. Given their recommendation, which contradicts INVEST, and their failure to address the limitations in their approach, it should be considered with caution that leaving out acceptance criteria might not be effective in actual ASD settings.

While previously discussed studies often lack concrete and usable best practices for Agile implementation, the research by Nisyak et al. (2020) provides clear strategies for applying the INVEST framework. Specifically, the strategies provided have been applied in government outsourcing projects. While our particular interest lies with INVEST, it is important to note that this study exceeds those discussions. Nisyak et al. (2020) suggest using the DSDM method, which stands for Dynamic Systems Development Method, to better manage requirements. Their work also recommends using the RE-KOMBINE framework for a detailed analysis of requirements and suggests applying the INVEST criteria thereafter for handling big user stories. They also highlight the importance of including scheduling buffers and encourages strong communication practices like regular stakeholder meetings and having an on-site customer. The latter strategies are specifically designed to address the challenges in government projects, focusing on adaptability and effective communication in Agile environments (Nisyak et al., 2020). However, the study's findings, based on Indonesian government projects, may not directly translate to ASD teams in the Netherlands, considering the possible differences in organizational cultures and project management approaches. Yet, it does show that considering on-site or remote meetings could potentially make a difference in a project's success.

Offering a unique perspective in this literature review, the study by Poth et al. (2019) directly addresses Agile adoption in large enterprises. It emphasizes that tailored approaches, rather than a one-size-fits-all approach will be more effectively implemented. They highlight the necessity of adapting Agile methodologies to suit the unique needs and challenges of large, diverse organizations. The study proposes a transition kit, integrating tools like INVEST, Scrum, and Kanban, and stresses the importance of

coaching and governance (Poth et al., 2019). This approach highlights the need to adjust Agile strategies to fit the unique team behaviors and culture in large companies.

With the literature reviewed in this subchapter, we looked into how the INVEST framework is applied in ASD, based on both theory and real-world case studies. This analysis responds to **SRQ 3**, which asks "What actionable insights can be drawn from the analysis of the INVEST framework's application in case studies?". The studies point out the need for balancing and adapting technical (Agile) methods with how unique teams interact and the culture of their organizations. Some studies suggest focusing more on the story part of user stories rather than detailed criteria, while others emphasize the need for good communication and flexibility. In general, these studies highlight the importance of a flexible approach in Agile practices, offering valuable insights on how to customize Agile strategies effectively for successful implementation in various types of organizations.

4. Experimental Procedure

Our study uses an embedded experimental-case study design to observe how INVEST can improve ASD. As Yin (2009) explains, embedded case study designs allow for the inclusion of multiple units of analysis within a single study, creating a better understanding of the phenomenon being studied. Within our context, this means we combine real-world projects with manipulating user stories to see how these changes affect the quality of work and the overall process. As such, we create a picture of INVEST's benefits in Agile projects, ensuring our findings are practical and based on evidence. This approach helps us understand not only if INVEST works in ASD, but also how and why. We explore a structure for the embedded design to complement the literature review on INVEST's effectiveness in ASD. This involves multiple cases to offer a diversified understanding of Agile practices and INVEST's application.

By examining a variety of projects, we gather a wider range of data, leading to more substantiated conclusions. Along with that, multiple case studies enable us to identify patterns, commonalities, and differences across various settings. In this study, we address each case individually, following the order presented in Subchapter 4.2. Within that order, we also introduce `SRQ 4`, `SRQ 5`, and `SRQ 6`. For each case, we answer these SRQs individually and end with exploring a comparison of results between cases.

4.1 Case Study Enrollment

Before case studies can take place, organizations must be contacted and requested to partake in the project. We used convenience sampling, a method where participants are selected based on their availability. In our case, we applied convenience sampling to recruit Agile-working project teams who were either unfamiliar with the framework or had not previously applied it in practice. This approach is efficient but may introduce bias, as it does not ensure a random sample from the target population. Within this experiment, we only consider projects based in the Netherlands. After project teams have agreed on partaking in the experiment, a consent form must be signed by the team members. This consent form is presented in Appendix A to ensure ethical standards and participant understanding of the study's scope, as well as rights regarding participation and data use.

We anonymize any sensitive information like product, software, team member, client or company names to ensure confidentiality and privacy for all parties involved. We specifically target cases where the teams have been collaborating for at least one month and are expected to continue for a minimum of a month at the start of the experiment. Depending on the Sprint length, which typically ranges from one to four weeks, this time frame allows us to capture at least one complete Sprint cycle "before and after" snapshot. That is, we include user stories from a period of time prior to experimental intervention and also user stories after experimental intervention. By (post-)monitoring these cycles, we aim to provide a clearer understanding of how INVEST influences project outcomes when applied over an extended period.

4.2 Case Study Procedure

In this subchapter, we discuss the procedure applied for each case study in this experiment. To apply this procedure systematically, we created Figure 5, which represents the flow of the case study procedure. It discusses four stages: Entry Interviews, Workshops, Evaluation, and Reflection. Each phase is broken down into activities further discussed in their respective descriptions.

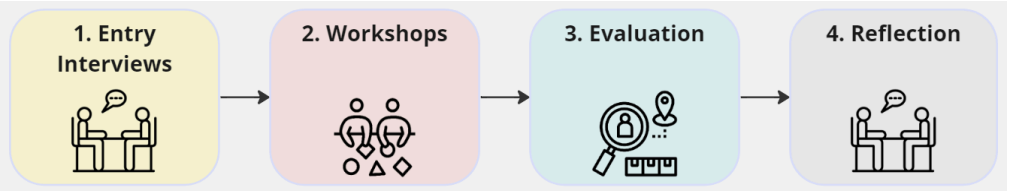

Figure 5: Research process diagram visualizing the four phases for each case study

4.2.1 Entry Interviews

The interviews are semi-structured and require the signing of the consent form before participation. Interviews are planned to last around thirty minutes with a maximum of an hour. The procedure for interviewing the Product Owner is described in Appendix B.1, while the procedure for Developers can be found in Appendix B.2. Although there is no requirement for pre-interview preparation by the interviewees, we will provide an overview of the INVEST framework to establish a common ground. The case study begins by conducting entry interviews with the PO and then a developer from the project team as shown in Figure 6.

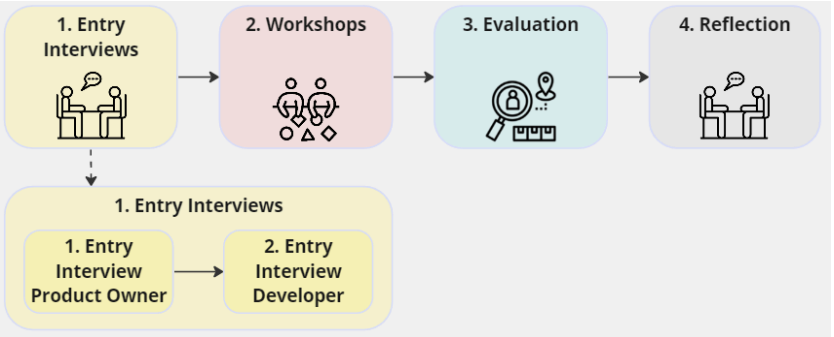

Figure 6: Phase 1 of the Research process diagram

The objective of these interviews is to gain generic knowledge about the company and project, including the industry they serve, the project's purpose, duration, and other contextual information. The interview will also explore the duration of their roles, the length of their involvement in the project, and their overall experience in Agile settings and the IT sector on a broader level. An essential aspect of this initial stage is to understand their awareness and usage of the INVEST framework in their current project. This forms the first stage of our embedded experimental-case study.

These interviews, along with subsequent activities, will be conducted online or in person, depending on what is suitable for the interviewees. To allow for documentation, these interviews will be recorded using generic voice recording software. After drafting the contexts for case studies, we follow up on these with the interviewees to ensure that the conversations held are accurately represented.

4.2.2 Workshops

The next phase involves three workshops, with each lasting for two to four hours, visualized Figure 7.

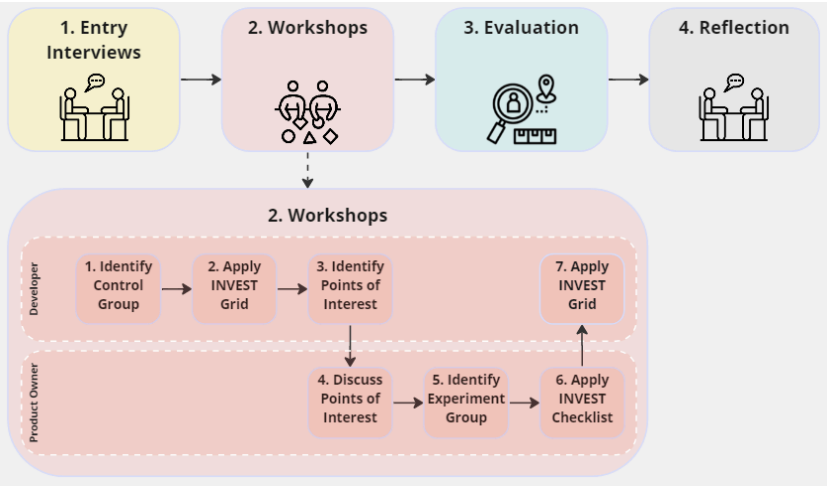

Figure 7: Phase 2 of the Research process diagram

For the first activity of these three, a workshop is carried out with the developer involved. The developer is requested to review the user stories they have completed in their past Sprints. For this review, the developer applies the INVEST grid as presented in Figure 1. After rating the user stories with the grid, we store the user stories along with any linked data such as their ratings, issues, discussions, labels, and estimates, labeling them the "Control group". The size of this group depends on the project dynamics; for instance, if there are multiple full-time developers, the experiment will span one or two sprints, whereas, for projects with fewer developers, or fewer hours of labor per sprint, it may extend over three or more sprints. This is done to achieve a comparable effort distribution and ensure a meaningful assessment. Note that this set of user stories only consists of developed user stories prior to the experiment. By only considering such user stories, we establish a reliable "before" snapshot as the control group where INVEST was not explicitly considered.

As such, we will be able to analyze the control group and assess to what extent the completed user stories already adhered to INVEST. During this workshop, we also focus on which criteria from INVEST are not cared for sufficiently. This is an analysis carried out using the ratings given by developers, and their expressed thoughts during the workshop. After the analysis, we aim to have a feedback loop with the developer to confirm the findings of the workshop. By doing so, this workshop helps answer our next sub-research question **SRQ 4**: "How do Agile teams perceive their user stories against the INVEST criteria prior to experimental intervention?" It should be noted that different projects may encounter different obstacles, but at the same time, we might uncover recurring issues. For instance, multiple cases, regardless of experience, might encounter difficulties with creating independent user stories.

Afterward, we conduct a workshop with the respective PO to enhance user stories that are written, but yet to be refined with the project team. These user stories form the experimental group of this study. The goal of the workshop is to collaboratively enhance user stories to align them with the INVEST criteria. The PO brings user stories from their respective project(s) to this workshop. First, we store the original content of these user stories. Appendix C introduces the INVEST Checklist to support the PO in assessing their user stories before refining them with their team members. This checklist is based on the findings and takeaways discussed in Chapter 3. While having the INVEST grid to assess user stories, this checklist complements the grid by not only offering a comprehensive description per criterion but also by asking relevant questions to critically assess each aspect of a user story. For a more targeted and effective application of the INVEST framework, we reflect on the results of the workshops with the developers. We put the main focus on the criteria that scored the lowest, going by the ratings the developers gave in the workshop prior to the one with the PO. Then, the enhanced user stories are saved and labeled as the experimental group. By storing both the original and reworked versions of these user stories, we ensure the traceability and reproducibility of the workshop's outcomes.

Following the enhancement of user stories from the experimental group, the next step involves discussing them with the development team in refinement meetings. This activity is required but not monitored, and is picked up by the respective PO. We reflect upon the refinement process through discussions with the participating developer. Once these user stories have been refined, the participating developer will be requested to evaluate the refined user stories using the INVEST grid, as this time the focus is on the experimental group. From here on, we can analyze the effectiveness of the INVEST checklist by comparing the scores of the control group to the scores of the experimental group. In this way, we will be able to understand if the checklist improves the adherence to INVEST. In case the developer gives low scores when applying the grid, we request the developer to elaborate on that decision. Together with the progress from previous workshops, this third workshop will help answer **SRQ 5**: "To what extent can the INVEST criteria be flexibly adapted to accommodate diverse project needs and team dynamics?"

4.2.3 Evaluation

Following the refinements, the refined user stories will be put onto the Product backlog - a repository where to-be-developed items are stored. With this experiment, we will observe over time how the developers work on these user stories. This phase starts once the user stories are included in their Sprint. Figure 8 illustrates our data collection process for quantitatively assessing progress.

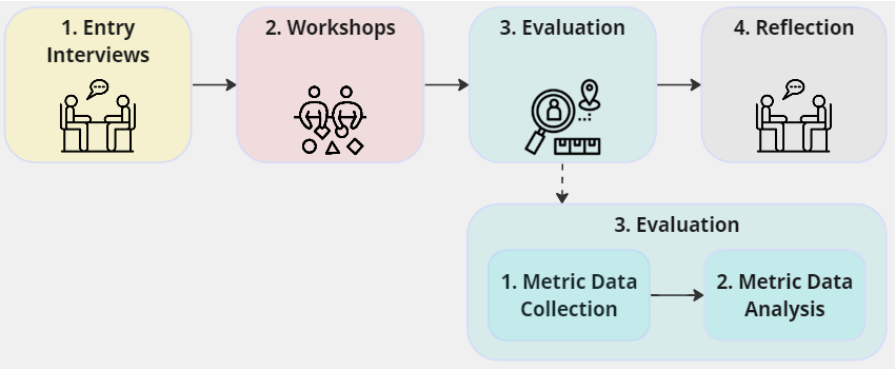

Figure 8: Phase 3 of the Research process diagram

In this process, we pay close attention to available metrics such as burndown. Additionally, we closely monitor issues raised, including details like their complexity or priority. An example of a significant observation would be if an INVEST-aligned story is developed in a day, whereas similar user stories typically require two days, indicating its effectiveness. A different example would be where a user story from the experimental group leads to no complex issues raised, while similar, control group-aligned user stories raise complex issues. To exclude coincidences, the participating developer is requested to express their chain of thoughts throughout the developments. By evaluating the complexity of issues raised, we can determine whether INVEST-aligned user stories help reduce complexity. This complexity will be determined by the developer on a scale of three levels: Simple, Normal, and Complex, along with substantiations.

4.2.4 Reflection

Once the user stories are developed and are considered “Done”, the study will gather subjective feedback from the developers regarding their experience working with INVEST-aligned user stories as shown in Figure 9.

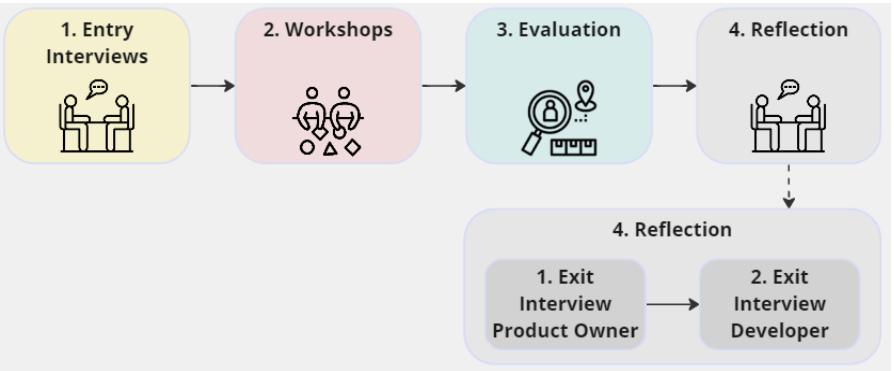

Figure 9: Phase 4 of the Research process diagram

In a reflection of approximately half an hour, we will focus on their preferences, any challenges faced, and their perceptions of the framework's effectiveness. Additionally, POs will be asked whether using the checklist made it easier to assess the quality of their user stories compared to their previous methods, if any. The procedure for both roles is nearly identical and is mentioned in Appendix B.3. Going by their experiences, we explore SRQ 6: “What are the key challenges, limitations, and benefits faced by Agile

teams in implementing the INVEST framework?" This SRQ is explored primarily through the insights from the reflection but also extends from the other activities in this experiment.

4.3 Comparative Analysis

In comparing case studies for this experiment, it is crucial to recognize that the diverse contexts of each project pose a notable challenge to direct comparisons. Each case study is unique, shaped by specific industry requirements, project goals, team structures, and experiences. This diversity impacts on how the INVEST framework can be applied in each case study. For instance, some teams might place more emphasis on the Small and Testable criteria of INVEST, while others may prioritize Independent and Negotiable, depending on their project's specific needs. Even when evaluating the same criteria, the degree of emphasis can vary significantly. These variations in the application of INVEST create substantial obstacles to comparing outcomes consistently across different Agile projects. The experiment's ability to adapt to these unique contexts adds layers of complexity to any comparative analysis. Consequently, the primary goal is not to draw direct comparisons between cases but rather to understand how the INVEST framework is adapted to suit unique project contexts.

5. Case Study A

5.1 Case Context

Case A explores a major international supermarket chain in over 30 countries, impacting 2700 employees. We interviewed a solution consultant (SC) and a developer from the Netherlands, both involved in Project A (year 2023) and Project B (year 2024). The organization uses low-code platforms to centralize recall information. The agile team included a lead, senior, intermediate, and three junior developers. The application took four months and 2300 development hours to complete. A switch to a new platform led to Project B, with a smaller team of four developers and a Quality Developer. Project B used two-week sprints, planning three and a half sprints and a four-week security test, totaling about 20 weeks.

Project A achieved 200 SPs per sprint, but the team expected 80 SPs for Project B. User story enhancement is ad-hoc, focusing on practical solutions over formal theories, with INVEST not considered before the experiment. The organization lacks a consistent user story approach, with quality interest spiking in new projects but fading quickly. Poor quality stories have previously caused developer frustration and slowed progress. The team expected the experiment to improve team dynamics.

Originally, Project A's extensive preparation minimized the need for later refinements, leading to fewer questions during the development phase. Following this experience, the team started Project B with a crucial backlog review to address gaps in context and including meetings with stakeholders. At last, the developer emphasized the need for time and continuous collaboration to implement INVEST effectively, noting its potential benefits for project clarity and efficiency. However, they acknowledged the significant investment required for a widespread adoption of the framework.

5.2 Workshops

5.2.1 Workshop 1: Control Group Analysis

In the first workshop, we rated 21 user stories, totaling 88 SPs, visualized in Table 1. During the scoring, the developer noted the efficiency of merging related functionalities into larger user stories, yet faced difficulties with the Independent criterion. They mentioned (quote) *"user stories often depend on others within a process flow"*.

Score \ Criterion	I	N	V	E	S	T
Score of 0	1	0	0	0	0	0
Score of 1	4	0	1	0	0	11
Score of 2	5	3	0	13	0	2
Score of 3	11	18	20	8	21	8

Table 1: INVEST Criteria Scoring Distribution for 21 User Stories, Case A - Project A

According to the developer, the Negotiable score depends on the author of the user story, with the SC being praised for their flexibility, pointing out their openness to suggestions. The Testable criterion mostly relied on implied guidelines, expected to be understood by developers or peer reviewers, which often resulted in incomplete acceptance tests. Based on the scores as presented in Table 1, we observed a skill in creating valuable and negotiable user stories, reflecting the team's understanding of end users' needs. Full scores for Small user stories show they are well-sized for iterative development.

To answer SRQ 4, we observed a variance in scores for Independent and Testable. Not only did dependencies cause bottlenecks and testing delays, we also found that while some stories were clear for development, some lacked detail for testing. The Estimable criterion often received a 2/3 score due to additionally required validations. Given the presence of several criteria with scores of 0 and 1, the main focus for the second workshop was not the Estimable criterion but rather the Independent and Testable criteria. Furthermore, the Negotiable, Valuable, and Small criteria appeared to be appropriately addressed.

5.2.2 Workshop 2: Experimental Group Enhancement

In our second workshop, we went through three steps: reviewing the first workshop's issues, evaluating new user stories, and enhancing those newly written user stories where deemed possible. Following Workshop 1, we specifically addressed the excessive splitting of user stories, leading to many interdependencies. We also discussed the need for writing clear acceptance criteria. While developers often know the testing approach, the SC acknowledged the importance of clear, testable user stories and the potential for varying interpretations of tests among team members and end users who test features.

In the workshop, we reviewed 22 randomly selected, yet to be developed user stories, assessing each against INVEST and marking each criterion with a check mark or cross, decided by the SC. They highlighted that 12 user stories lack in both the Valuable and Testable criteria. Despite challenges during Workshop 1, the Independent criterion showed improvement. The result of this evaluation is shown in Table 2, where we see an overview of the different combinations of INVEST, as assessed by the SC.

Combination I N V E S T	Frequency
✓ ✓ ✗ ✓ ✓ ✗	12
✓ ✓ ✓ ✓ ✓ ✗	6
✗ ✓ ✗ ✓ ✓ ✗	2
✗ ✓ ✓ ✗ ✓ ✗	1
✓ ✓ ✓ ✗ ✓ ✓	1

✓ Cared for sufficiently
✗ Not cared for

Table 2: INVEST Criteria Scoring Distribution for 21 User Stories, Case A - Project B

When enhancing the user stories, we observed that the SC uses artificial intelligence (AI) for writing user stories, apparently reaching up to 80% usage in some of their projects. AI would suggest acceptance criteria which were then refined before inclusion, proving particularly useful for business users involved in testing. The lack of clear value was addressed by adding more context at the beginning of the user story, explaining why the specified solution is required to reach the desired functionality. Also, the SC believed that using the framework sooner would have further improved the user stories and also saved time. They believe this approach will make stories more valuable and testable, reducing future issues.

5.2.3 Workshop 3: Experimental Group Evaluation

In Workshop 3, we collaborated with the developer to score the 22 user stories from Workshop 2. The results, shown in Table 3, indicate an improvement compared to the 21 user stories from Workshop 1. All user stories scored either a 2/3 or 3/3 on the INVEST criteria, with no scores of 0/3 or 1/3, except for Independent. This suggested that Workshop 2's enhancements led to better-aligned user stories.

Score \ Criterion	I	N	V	E	S	T
Score of 0	0	0	0	0	0	0
Score of 1	2	0	0	0	0	0
Score of 2	7	1	3	5	0	1
Score of 3	13	21	19	17	22	21

Table 3: INVEST Criteria Scoring Distribution for 22 User Stories, Case A - Project B

The Independent criterion improved, with 13 user stories receiving a score of 3/3, though a score of 1/3 was still observed twice, highlighting the challenge of achieving user story independence. Ten of the 22 user stories received a score of 3/3 across all six criteria. The Negotiable criterion scored a 3/3 for 21/22 user stories, demonstrating the team's ability to create adaptable user stories, and the Valuable criterion also excelled, with 18 user stories scoring a 3/3. This confirms Workshop 2's enhancements made user stories more relevant, though there was no significant improvement compared to the control group.

For Estimable, the scores indicate that developers find it easier to estimate the effort required for the user stories, with 17 user stories scoring a 3/3. This improvement may lead to more accurate sprint planning and resource allocation. Similar to Workshops 1 and 2, the Small criterion scored perfectly, with all 22 user stories receiving a score of 3/3. Lastly, the Testable criterion, which previously saw the most significant need for enhancement, had 21/22 user stories with a score of 3/3. This increase suggests that the acceptance criteria are now clearer, leading to more effective testing and quality assurance activities.

The findings from Workshop 3 demonstrate the adaptability of the INVEST criteria, addressing **SRQ 5**. The improved scores for Project B indicate effective adjustment to project needs and team dynamics, with a shift toward a structured use of INVEST principles. The initial lack of uniformity in writing user stories has moved toward a more structured method, enhancing team dynamics by reducing ambiguity, increasing clarity, and aligning expectations. This underscores the importance of collaboration and time investment between the SC and developers to implement INVEST effectively. However, dependencies between user stories still need improvement.

5.3 Reflection

Following the workshops, we continued with reflecting on both the study and INVEST criteria with both participants. Besides finding the experiment insightful, the SC appreciated the focus on their own user stories. They found that expressing the value more clearly had great effects on the developers' understanding, and noted that while Testable was also of significant value, it often lagged due to the effort required. They valued Estimable but found Independent challenging due to dependencies from splitting larger stories. They proposed a VEST framework, excluding Independent and Negotiable. The study improved their test policy with clear guidelines for business users. Also, the SC recommended INVEST for future projects, emphasizing the need for methods to better incorporate it into daily work.

Reflecting on the experiment with the developer, we found that applying the framework to user stories was initially time-consuming and challenging, but it became easier with practice. Although low scores did not always indicate problems, the framework helped identify areas for improvement. Putting emphasis on criteria like Independent and Testable led to better clarity and more complete acceptance criteria. However, they did not notice a significant change in development or communication efficiency. They considered the framework especially useful for complex projects, and some criteria, like Small, were seen as unnecessary. They explained that the experiment showed them that using INVEST as a guideline, rather than a strict rule, would help maintain certain standards of user story quality.

Addressing **SRQ 6**, the team faced challenges with INVEST, notably in maintaining user story independence and the significant effort required to make stories testable. Some criteria, like Negotiable and Small, add little value, and applying the framework initially is considered time-consuming. However, benefits include clearer value expression, better acceptance criteria, and improved test policies. The framework helps maintain user story quality, with the PO suggesting adapting the framework to a VEST model, excluding Independent and Negotiable, to better suit their specific needs.

6. Case Study B

6.1 Case Context

For Case B, we study a Dutch fertilizer company that has been in business for over 100 years, making an impact worldwide. The project we focus on in this study aims at streamlining and standardizing their sales team's work with regards to pricing, stock levels, and shipping, named the sales tool, which is a low-code application. We spoke to the PO, who has been with the company for 12 years, managing the supply chain previously. They switched roles looking for new challenges and to apply their company's knowledge more effectively. This shift meant focusing on planning rather than reacting to daily issues. Meanwhile, the sales tool involved overcoming initial resistance and adapting to a new way of working, in which the support of their five regional directors was crucial, getting the approximately 20 sales managers on board.

The PO quickly learned their role with a two-day preparation, all while facing challenges with user story formats and aligning them with business needs. They struggled to extract detailed feedback, often missing critical information. Skeptical of frameworks derived from academia, they emphasized the need for clear, concise user stories that initiate conversation and fit the project's style. Open to new approaches, the PO aims to improve their project management by adapting and learning from these experiences.

Besides the PO, we spoke to the externally hired senior developer handling this project. The developer began their IT career in low-code development in 2019 and expanded their role in multiple projects since joining the sales tool project mid-2022. Together with an intermediate developer, they took over the work from another senior developer at the time. They appreciate the comprehensive Scrum setup in the project, a notable improvement from previous experiences where they handled extra tasks like writing user stories and creating mockups, affecting the workload. Despite some planning challenges, the team's openness to trying new frameworks like INVEST, along with the PO's learning attitude and openness to suggestions, highlights a culture of continuous learning. This is also substantiated by the developer who mentioned that costing time might be problematic, but as long as they are on the same page, it will not be a big issue.

6.2 Workshops

6.2.1 Workshop 1: Control Group Analysis

In our first workshop, we assessed 17 user stories selected on recency. The developer expressed their preference for detailed user stories to avoid rework and make discussions easier. Another reason for rework is that the user stories often overlook the business value, visible in the low scores for Valuable in Table 4.

Score \ Criterion	I	N	V	E	S	T
Score of 0	2	0	5	3	0	2
Score of 1	0	1	7	7	2	15
Score of 2	6	0	1	5	5	0
Score of 3	9	16	4	2	10	0

Table 4: INVEST Criteria Scoring Distribution for 17 User Stories, Case B

While demoing recent developments, a business user even found a feature limiting, highlighting the need for more user feedback before developments take place. Nevertheless, the development team is still encouraged to offer suggestions and enhancements, increasing the scores for Negotiable. The developer found that user stories were often difficult to estimate, suggesting splitting larger stories into smaller, manageable parts for clarity and easier assignment. Although the dependencies would increase, the developer found the Independent criterion to not be a priority with (quote) *"more pressing issues to tackle"*.

The developer believes that applying the Connextra template can significantly enhance the Valuable criterion. In the context of some epics, discussions often occur early in the process. However, when new

team members join later, they lack this important information. This gap of knowledge often leads to challenges in estimability; while new colleagues may seek answers, their limited understanding of the business restricts them. Writing smaller user stories could mitigate this issue by allowing for more detailed explanations. Furthermore, the process of testing presents its own challenges, largely because formal acceptance tests are usually absent. This lack means that development and testing can proceed based on one set of expectations, only for the checks by the PO to introduce entirely different tests.

Following SRQ 4, the main observation was a lack of complete user stories. They were found to often be incomplete in terms of described functionality, their added value, and their testing strategies. We figured that this issue could be tackled by vertically splitting the user stories to limit the number of functionalities taken care of in a user story, and then providing expanded descriptions per user story. These descriptions should not only better cover the functionalities, but also the business value and testing techniques. Therefore, Workshop 2 is expected to tackle the Valuable, Estimable, Small and Testable criteria.

6.2.1 Workshop 2: Experimental Group Enhancement

We initiated the workshop by going over the INVEST grid with the PO, and discussing the results that we observed with the developer during Workshop 1. This workshop included eight user stories, as seen in Table 5. One of the eight user stories was considered too large, and was set to be split into seven different user stories after a collective refinement session, making it a total of 14 user stories for Workshop 2.

Combination I N V E S T	Frequency
✓ ✓ ✗ ✓ ✓ ✗	4
✓ ✓ ✓ ✓ ✓ ✗	1
✗ ✓ ✗ ✗ ✓ ✗	1
✓ ✗ ✗ ✗ ✗ ✓	1
✓ ✗ ✗ ✓ ✓ ✓	1

✓ Cared for sufficiently
✗ Not cared for

Table 5: INVEST Criteria Scoring Distribution for 8 User Stories, Case B

The contents of these user stories were mainly focusing on enhancements for existing functionalities, as the project has been in its final stages since early 2024. As a result, the enhancements are mainly end-user requests, and therefore not always as negotiable. During the workshop, the PO was not always receptive to modifying the user story, often expressing skepticism (quote): *"if we could read the user story and understand everything, why would we need a refinement?"* also noting that the application of INVEST would be more valuable for a new project, stating that concepts would then require more explanations.

From the workshop, we figured that four of the eight discussed backlog items lacked both expressed value and testability, as shown in Table 5. Regarding the value, the PO finds it impossible to clearly note down the added value of certain functionalities, as their business is considered highly complex. The value is therefore discussed during the refinement session(s), rather than textually within the user story. Regarding testability, the PO considers it the responsibility of the tester to come up with test strategies or test plans. They find that the acceptance criteria should be sufficient to know how to test the user story, and that figuring out any additional edge cases is where the tester adds value to the development process. As a result, the PO did not add additional test plans besides the acceptance criteria for any user story.

6.2.3 Workshop 3: Experimental Group Evaluation

With the third workshop, we reflected with the developer on what was enhanced during the previous workshop. The single large user story was also split up beforehand, giving us 14 user stories to review. The scores for these 14 user stories are visualized in Table 6. For 9/14 of these user stories, all criteria received the maximum score, whereas 0/8 from Workshop 1 received the maximum score for all criteria.

Score \ Criterion	I	N	V	E	S	T
Score of 0	0	1	0	1	0	0
Score of 1	0	1	1	0	0	0
Score of 2	0	1	0	0	0	0
Score of 3	14	11	13	13	14	14

Table 6: INVEST Criteria Scoring Distribution for 14 User Stories, Case B

During the rating of these improved user stories, the developer mentioned they had discussed the user stories with the intermediate developer beforehand. They had concluded that these enhanced user stories have improved significantly and cannot be compared to the former style. After assigning scores for all 14 user stories, we found that the former user stories, as can be seen in Table 4, are of no match against the scores presented in Table 6 when merely comparing scores. The developer pointed out during our discussion that the scores likely improved due to the PO's openness and their repeated assurances of being available for improvements, which they knew would boost the team's efficiency over time.

In addressing SRQ 5, it became clear from Workshop 3 that INVEST can be flexibly adapted within this project and team. The large improvements in the user stories from Workshop 1 to Workshop 3, where the scores for 9 out of 14 user stories received maximum ratings compared to none previously, substantiate this adaptability.

6.3 Reflection

When reviewing the study results, the PO described the experience as mixed but insightful. They found criteria like Valuable and Testable beneficial, noting a significant increase in their understanding of INVEST. However, they expressed skepticism about some criteria, feeling they were redundant or overly challenging, such as breaking down user stories into very small parts, which could lead to dependencies. They observed that Valuable and Small are intertwined, as summarizing value becomes easier with smaller user stories. The PO appreciated the structured reflection and became more aware of audience demands. They noted the required level of detail depends on team dynamics and developer freedom, finding the INVEST checklist too general for this purpose. Despite a better understanding on INVEST, they felt the benefits were limited and unlikely to lead to long-term changes without ongoing training.

The developer, on the other hand, was left with no mixed experience as they found the study had greatly affected their user stories. They noted improvements in the clarity, value, and testability of user stories, which made development more efficient and required fewer clarifications during the sprint in which most of the 14 user stories were picked up. The developer found the Valuable and Testable criteria most useful, while Negotiable and Estimable were more challenging to assess, particularly for technical tasks like APIs where negotiation is limited. Despite some initial fatigue with revisiting old user stories, they appreciated the overall framework and the support received, recognizing it led to better project quality and smoother refinement sessions. In other words, the user stories not only adhered better to the INVEST criteria but were also easier to work with. They endorsed the use of INVEST for future projects, highlighting its positive impact on the team's efficiency and understanding of user stories.

The team encountered several challenges and limitations, yet also discovered notable benefits by committing to the experiment. To answer SRQ 6, from the developer's perspective, a key challenge is ensuring user stories meet all criteria, especially being both negotiable and estimable, which can be difficult for technical components like APIs where flexibility is limited. The PO highlighted difficulties in summarizing complex stories concisely and managing dependencies when breaking stories into smaller units. Despite these hurdles, the benefits include clearer, more valuable, and testable user stories, leading to smoother refinement sessions and reduced questions during sprints. As such, the developer found we enhanced their efficiency. The framework provides structure, enhancing user story quality, clarity, and team collaboration, yet requires improved guidelines.

7. Case Study C

7.1 Case Context

We discuss a software development company's shift from the waterfall to Agile methodology, focusing on projects aimed at both municipalities and improving healthcare coordination since 1996. The main focus for this study is on user stories for centralizing data for healthcare providers to enhance patient care. While transitioning to Agile, the team experiments with different techniques to ease their tasks. The development team, consisting of 3 Front-end (FE) developers and 3 Back-end (BE) developers, initially logged FE and BE tasks separately but considered them dependent. These tasks are slowly transitioned into unified user stories with subtasks, marking a shift in their workflow.

Despite the absence of a dedicated tester, they benefit from customers, usually functional managers, carrying out testing activities. These managers are responsible for translating user needs into technical requirements, and the PO turns these technical requirements into user stories for the development team.

The PO, with 1.5 years of experience in their role, aims to write clearer, independent user stories. They use prototypes to guide developments and participate in workshops to improve Agile methods, demonstrating a commitment to continuous learning. The developer, with nearly a year in their current role and four years of Scrum experience, is specialized in a FE framework. Having transitioned from a non-IT background to programming, they value Agile's blend of theory and flexibility. Their adaptability in handling both FE and BE tasks shows their versatility and growth-focused mindset.

7.2 Workshops

7.2.1 Workshop 1: Control Group Analysis

For the analysis of the pre-experimental state of INVEST in the project, together with the developer we went over 14 completed user stories, with the scores shown in Table 7. The set of user stories that were reviewed were chosen for their relevance. That is, we focused on including user stories that contained both the FE and BE activities within the user stories as separate tasks, rather than separate user stories.

Score \ Criterion	I	N	V	E	S	T
Score of 0	0	2	6	5	1	3
Score of 1	1	2	2	3	0	7
Score of 2	5	3	2	1	1	0
Score of 3	8	7	4	5	12	4

Table 7: INVEST Criteria Scoring Distribution for 14 User Stories, Case C

With eight user stories receiving a 3/3 score for Independent and 12 user stories receiving that maximum score for Small, these criteria are considered best cared for within this project, indicating well-managed dependencies and well-sized user stories. However, user stories frequently lacked detailed or proper descriptions and alternative solutions would not be considered. Along with often missing the value or context to explain end-users' needs, these issues led to low scores for Negotiable and Valuable, as shown in Table 7. Where descriptions were considered lacking, we figured that user stories were difficult to estimate, showing that these criteria are intertwined. The low scores for Estimable were unsurprising, as it was well-known within the team that estimations were often inaccurate and developments took longer to complete than initially planned. At last, we found that the user stories often had acceptance criteria indications, but generally-speaking, these were considered incomplete.

Considering SRQ 4, the developer viewed their user stories as independent and small, yet found that they are lacking in being negotiable, valuable, estimable, and testable prior to experimental intervention. With regards to Workshop 2, the developer mentioned (quote) *"a more detailed user story allows for deeper consideration of aspects that often require thought, enabling one to address these issues effectively."*

7.2.1 Workshop 2: Experimental Group Enhancement

We initiated this workshop by reflecting on the previous results. By discussing what criteria generally require improvements, and going over both the INVEST grid and checklist, we were able to reach a consensus on how to assess which of the 12 user stories, shown in Table 8, need which improvements.

Combination I N V E S T	Frequency
✓ ✓ ✗ ✓ ✓ ✗	3
✓ ✓ ✓ ✗ ✓ ✗	2
✓ ✓ ✓ ✓ ✓ ✓	1
✗ ✓ ✓ ✓ ✓ ✓	1
✓ ✓ ✗ ✓ ✓ ✓	1
✓ ✓ ✗ ✗ ✓ ✗	1
✓ ✗ ✗ ✓ ✓ ✗	1
✓ ✓ ✗ ✗ ✗ ✗	1
✗ ✗ ✗ ✗ ✓ ✗	1

✓ Cared for sufficiently
✗ Not cared for

Table 8: INVEST Criteria Scoring Distribution for 12 User Stories, Case C

Regarding the writing of user stories, the PO elaborated on a template they apply that enforces a "given-when-then" format, in Dutch, to ensure clarity and consistency across user stories. This structured template is accompanied by acceptance criteria and additional information to support the team.

The PO mentioned that the FE developers are ahead in their work, while the BE developers are behind. However, the BE developers are aware of what needs to be done, something the PO cannot detail for them as effectively as for the FE developers. While going over user stories during the workshop, the PO would occasionally show the corresponding designs to create a better understanding of its contents. Designs were also linked within most of the user stories, to achieve that same goal with the development team.

While using the Connextra template, the benefits were more often than not missing context. As can be found in Table 8, for 8/12 user stories, the value described initially was not sufficient, according to the PO. Along with Testable, which had to be addressed 9/12 times, these two criteria caused the most rework during the workshop. The acceptance criteria often only addressed the happy flow, and while unhappy flows would be tested by the PO, they were generally excluded from the user stories. We found that some user stories are still dependent on others, and we discovered that addressing these dependencies, for example by merging the user stories, is not suitable.

7.2.3 Workshop 3: Experimental Group Evaluation

Following Workshop 1, we identified Negotiable, Valuable, Estimable, and Testable as points of improvement by analyzing 14 user stories. The results of the workshop are visualized in Table 9.

Criterion / Score	I	N	V	E	S	T
Score of 0	2	1	0	2	0	1
Score of 1	0	1	2	3	0	0
Score of 2	1	3	1	1	1	2
Score of 3	9	7	9	6	11	9

Table 9: INVEST Criteria Scoring Distribution for 12 User Stories, Case C

While carrying out improvements for 12 new user stories during Workshop 2, we prioritized the Valuable and Testable criteria after finding that they were not cared for within 8/12 and 9/12 user stories. For Workshop 3, we identified that Valuable and Testable had improved most: previously, during Workshop 1,

the score of 3/3 was assigned 4/14 times to both criteria, but after experimental intervention, both criteria received the maximum score 9/12 times.

While not identifying considerable improvements for Independent, Negotiable, and Small, we see slightly improved scores for Estimable. This suggests better descriptions which allow the developers to estimate their efforts more accurately. For 5/12 user stories, a score of 3/3 was given to all six criteria. This improvement confirms that the enhancements of Workshop 2 have ensured the user stories are considered more valuable after experimental intervention, albeit showing no significant improvement compared to the control group.

The findings from Workshop 3 show that within this project, certain INVEST criteria are adaptable. To address **SRQ 5**, the score improvements for Case C show effective adjustments to the Valuable and Testable criteria, with slight improvements for its estimability. Although the user stories initially lacked on the areas of Independent and Negotiable as well, these criteria do not show significant improvements.

7.3 Reflection

The reflections from the PO and the developer offered insights into the use of the INVEST framework within their Agile team. The PO mentioned that (quote) *"INVEST helps ensure you stay focused on writing effective user stories"*, noting improvements in the Valuable and Testable criteria after workshops. They acknowledged that while FE developers benefit significantly from INVEST, the BE developers, who are more used to waterfall methodologies, find it less intuitive. They also highlighted the importance of understanding the rationale behind stories to avoid misalignments during development. The PO even stated that they plan to make sure other project teams at their company adopt the INVEST criteria.

The developer highlighted how applying INVEST has enhanced the clarity and value of user stories, appreciating its role in clarifying the purpose and benefits of features for better solutions. However, they noted that Negotiable and Estimable remain difficult to apply consistently. Balancing detail and developer input is crucial; strict requirements are needed for design elements, while functional aspects should allow for creativity. Also, the varying level of detail per story makes it hard to consistently assign a score. Additionally, Independent and Small are straightforward and often redundant. They suggested that while INVEST improves understanding, it must avoid being too prescriptive to maintain developer creativity.

Focusing on **SRQ 6**, we figured that Agile teams implementing the INVEST framework face several key challenges and limitations. One challenge is ensuring that all team members, regardless of their background (e.g., FE vs. BE developers), find the framework equally useful. Additionally, consistently applying criteria like Negotiable and Estimable can be difficult. Another limitation is the potential increase in time required to write detailed user stories, which can add administrative burden and slow down the development process. Despite these challenges, the benefits are significant. The framework helps create clearer, more valuable, and testable user stories. By ensuring user stories are independent, small, and well-defined, INVEST enables more manageable development tasks. This results in higher quality outputs and better alignment with user needs, with both the PO and developer noting improved clarity and value in stories, which contributes to more effective and efficient project execution.

8. Case Study D

8.1 Case Context

This project, initiated by a consultancy firm specializing in insurance, represents a collaborative effort to create a claims portal for managing absence notifications and policies. Low-code developments started in the year 2020, and seek to digitalize back-office systems, by reducing manual tasks through automation. Over time, the project saw transitions between different development teams, with the current team starting early 2023. With different shareholders, the project effectively functions as an innovative startup.

The low-code application is split into four subdomains, with a PO for each. Alongside, each subdomain has its lead developer carrying out cross-domain peer reviews. Together with a separate tester and architect overseeing all developments, the team works in two-week sprints, favoring the SMART criteria for writing its acceptance tests. For this case, we interviewed their developer with significant domain knowledge. They noted that user story quality is affected by POs' domain expertise rather than IT knowledge, but the team is open to trying new formats given they are not too time-consuming.

We also interviewed a person who holds dual roles as both Product Manager (PM) and member of the Board of Directors. As a PO at their previous job, a major insurance company, they transformed the IT landscape and implemented the SAFe methodology. Now, at a smaller company, their role has grown to include financial oversight and strategic decision-making. Despite those previous experiences, the current project applies standard templates for user stories without adhering to a specific methodology. The PM believes their current user story format is workable and does not need major changes, but sees room for improvement in content writing to maximize efficiency with minimal effort. They want to ensure that the PO or analyst does not adhere to a strict routine, to keep within capacity limits.

8.2 Workshops

8.2.1 Workshop 1: Control Group Analysis

In the assessment of the current situation, the developer and we collectively reviewed 11 user stories written by the PM, excluding those from other subdomains. The scores are visualized in Table 10.

Criterion / Score	I	N	V	E	S	T
Score of 0	4	2	2	1	0	0
Score of 1	1	2	3	2	0	6
Score of 2	2	4	2	2	1	2
Score of 3	4	3	4	6	10	3

Table 10: INVEST Criteria Scoring Distribution for 11 User Stories, Case D

The developer praised the open-ended nature of these user stories, allowing discussions and refinements. It is understood that the PM focuses on mapping approximately 80% of the functionality, leaving the remaining portion to the developers. The developer pointed out that, often, the background information was lacking, and it was not apparent how the functionality of the user story would satisfy the given need.

It was also observed that although many user stories had acceptance criteria, they mainly covered the "happy flow" and overlooked other scenarios like the "unhappy path", previously resulting in testing gaps. The developer mentioned the difficulty in writing independent user stories, as they tend to describe step-by-step processes. The main goal was to improve the Valuable and Testable criteria, with a secondary emphasis on making them Negotiable and Estimable, before focusing on the Independent criterion.

Table 10 indicates that the user stories, specifically for SRQ 4, require improvements in all areas except size to appropriately meet the INVEST criteria. Before any experimental intervention, many of the user stories are seen as too dependent, not flexible, unclear in their value, difficult to estimate, and hard to test.

8.2.1 Workshop 2: Experimental Group Enhancement

To kick off the second workshop for the insurance project, together with the PM we went over the results of the previous workshop. We discussed that the top priorities were making the user stories Valuable and Testable, with the developer noting that the focus was on improving these areas first. When going over six newly created user stories, of which the results can be found in Table 11, the PM would present the relevant application designs to give more context before making adjustments.

Combination I N V E S T	Frequency
✓ ✓ ✗ ✓ ✓ ✓	3
✓ ✓ ✓ ✓ ✓ ✓	2
✗ ✓ ✗ ✓ ✓ ✗	1

✓ Cared for sufficiently
✗ Not cared for

Table 11: INVEST Criteria Scoring Distribution for 6 User Stories, Case D

When discussing the INVEST criteria, the PM would disagree on determining the independence of user stories. They would mention that every user story would require some kind of previous functionality to be implemented before being able to start working on the next one, making each and every user story in some way dependent on a previous user story. As such, we found that Independent would be a criterion to disregard when applying a framework to improve user story quality.

At times, the PM would state that the acceptance criteria seem to be incomplete, but would not know additional criteria to add right away. Because of that, the testability would not always be sufficient, but would not be improved either. With three user stories scoring almost flawlessly and two receiving a perfect score, the PM dedicated these high scores to the way of working of their current company, which organizes training sessions for both their POs and PMs.

8.2.3 Workshop 3: Experimental Group Evaluation

Before the third workshop took place, the initial developer received the news of being reassigned to a different project. Afterwards, we onboarded a new developer from the same team to continue our research. To be able to resume from where we left off, we began the workshop by discussing the project goals and reviewing outcomes from previous workshops.

The new developer noted that user stories are linked in the project management tool, and if no links exist, a story is deemed independent. However, “is blocked by” relationships were sometimes incorrect, merely indicating that stories were related and needed simultaneous release. Despite a smaller sample size, the Independent criterion showed significant improvements over the control group, as seen in Table 12.

Score \ Criterion	I	N	V	E	S	T
Score of 0	1	0	0	0	0	0
Score of 1	1	0	0	0	0	2
Score of 2	0	1	0	1	0	1
Score of 3	4	5	6	5	6	3

Table 12: INVEST Criteria Scoring Distribution for 6 User Stories, Case D

The Negotiable and Estimable criteria improved, with five out of six user stories scoring 3/3 and one scoring 2/3 for both criteria. Valuable and Small achieved perfect 3/3 scores for all stories. Despite being a focus after Workshop 1, the Testable criterion only slightly improved, with two stories still scoring 1/3.

Answering SRQ 5, the application of the INVEST criteria to enhance user story quality showed notable improvements in the Independent, Negotiable, Valuable, and Estimable criteria. The Small criterion, as

before, still received the fullest scores. Despite focused efforts, the Testable criterion showed only slight improvement, suggesting that there is still room for further enhancement.

8.3 Reflection

With the originally onboarded developer having been reassigned to a different project, this reflection only took place with the PM, who found the study insightful. The PM noted how the quality of user stories improved over time. Initially, most criteria needed improvement except for Small. Workshops 2 and 3 showed significant progress, though efforts by the team, prior to the experiment, might have influenced these results. Challenges remained with Independent and Testable criteria, as we observed that further improvements could be made.

The PM appreciated INVEST for creating a common ground and enhancing quality, despite finding Independent and Negotiable less practical. They mentioned that rather than minimizing dependencies, they should be clearly defined. The PM found Valuable a useful addition, considering it more often when writing user stories. They viewed frameworks as flexible tools rather than strict rules and pointed out that INVEST emphasizes theory more than practical application. The research led to useful self-reflection, but the PM suggested that future studies should place a greater emphasis on quantitative analysis to achieve long-term benefits.

To discuss SRQ 6, the key challenges faced by the team in implementing INVEST included dealing with dependencies and handling the Negotiable criterion, which needed to be flexible but within the boundaries or scope of the user story. The limitations of the framework were seen in its theoretical nature, which did not always translate well to practical, real-world scenarios. However, the benefits included a significant improvement in the quality of use stories, especially in making them more clear for developers, helping their thought processes. The criteria served as a useful reminder and guide for maintaining quality standards, even if the exact criteria needed some adaptation for practicality.

9. Case Study E

9.1 Case Context

The developments that are discussed for Case E, are carried out by a company specializing in renewable energy solutions for the business sector. They are dedicated to continuously enhance the internal software systems to improve customer engagement and compliance with changing regulations in the Netherlands. Being part of a rebranding initiative by its international parent company, they experiment with ASD methodologies to adapt to the evolving business and regulations, showcasing their operational flexibility.

Their ecosystem architecture consists of several systems including contract management and customer self-service portals, managed by a PO with previous experiences as a test engineer. Working in sprints of three weeks, their team consists of the PO and four low-code developers who manage all developments and testing internally, supported by a functional specialist focusing on the data model. Along with the PO, their lead developer will support the study. With extensive experience in low-code programming, testing, and UI/UX design, we foresee to gain valuable, diverse perspectives that will enhance our findings.

An interview with the product owner, who oversees the project's strategic direction, showed that the team makes use of a Definition of Ready (DoR), which is based on INVEST. More specifically, it focuses on ensuring compliance with legal and privacy requirements, clear understanding and agreement, appropriately detailed acceptance criteria, minimized dependencies, accurately sized user stories, and achievable within a single sprint. Interestingly, although the DoR emphasizes the importance of INVEST, the PO mentions they do not make use of writing acceptance tests, leaving potential for differing expectations. The PO expects to see improvements for the Valuable and Testable criteria. Additionally, insights from the developer revealed that a section named "Points for attention" is often implemented, even though it mostly contains duplicate information when compared with the user story description.

9.2 Workshops

9.2.1 Workshop 1: Control Group Analysis

Prior to the workshop, we went over the user story format that is being used within the project. Given the regulatory requirements, each story begins with a mention of GDPR compliance or other relevant considerations. Followed by the Connextra template, preconditions are also noted where necessary, leading into the acceptance criteria. Following this, we address what the lead developer considers redundant points of interest, provide guidance on demonstrating the feature, including a URL for the screen currently in development, and conclude with technical notes that follow from refinements.

During the workshop, we analyzed 11 user stories, with the results presented in Table 13. We observed that for all criteria, except for Estimable and Testable, at least 9/11 stories received a 3/3 score. Consequently, we decided that those two criteria are the points of attention for the second workshop. Notably, the Small criterion scored a 3/3 for all 11 user stories indicating they are well-sized.

Score \ Criterion	I	N	V	E	S	T
Score of 0	2	0	0	1	0	0
Score of 1	0	1	1	2	0	3
Score of 2	0	1	1	3	0	1
Score of 3	9	9	9	5	11	7

Table 13: INVEST Criteria Scoring Distribution for 11 User Stories, Case E

Regarding SRQ 4, we find that the user stories in this agile team are perceived to be well-sized, with explicit mention of the added value, whilst being considered negotiable. Usually, these user stories can be picked up and delivered independently of others. However, the user stories are often hard to estimate due to a lack of comprehensive elaborations and difficult to test because of a lack of complete acceptance tests.

9.2.2 Workshop 2: Experimental Group Enhancement

Together with the PO, we began the workshop by reflecting on the first workshop. The developer had scored 11 user stories and prioritized the Estimable and Testable criteria. This second workshop focused on eight user stories, as shown in Table 14.

Combination I N V E S T	Frequency
✓ ✓ ✓ ✓ ✓ ✓	3
✗ ✓ ✗ ✓ ✓ ✗	1
✗ ✗ ✓ ✓ ✓ ✓	1
✓ ✓ ✗ ✗ ✓ ✓	1
✗ ✓ ✓ ✓ ✓ ✓	1
✓ ✗ ✓ ✓ ✓ ✓	1

✓ Cared for sufficiently
✗ Not cared for

Table 14: INVEST Criteria Scoring Distribution for 8 User Stories, Case E

When going over the user stories to define which criteria require more attention, from the perspective of the PO, we noticed that certain user stories rely on others to be completed first. To handle these dependencies, they typically address the first user story in one sprint, and then tackle the dependent user story in the following sprint. Additionally, not all acceptance criteria for these initial user stories were considered testable, as they serve as prerequisites for their dependent user stories to function. Furthermore, the PO mentioned that every user story is negotiable as long as proposals from the development team do not reduce the anticipated value for the customer.

With this activity, we found that three of the eight user stories scored sufficiently for all INVEST criteria. Three times, a user story was considered to be dependent, with two occurrences of having a user story that was not negotiable or valuable. As opposed to the developer's expectations, both the estimability and testability of the user stories were generally well addressed, with only one of the eight user stories lacking in both criteria. We reasoned that these high scores can likely be attributed to the fact that an extensive DoR has been implemented within the team, based on the INVEST criteria.

9.2.3 Workshop 3: Experimental Group Evaluation

Before the third workshop began, we learned that the developer we had initially onboarded would be taking an extended leave from their development tasks. In response, we introduced a new developer from the same team to continue our research efforts. To ensure the transition went seamless, we started the workshop by discussing the project goals and reviewing the achievements from previous workshops.

Following the scoring of the eight user stories, we find that the Independent criterion scores lower when compared to the user stories of the first workshop, as seen in Table 15. We noticed more "Is blocked by" relations within their project management tool, and even found that some user stories block each other, causing confusion for the developer.

Score \ Criterion	I	N	V	E	S	T
Score of 0	2	0	1	1	0	1
Score of 1	0	0	0	0	0	0
Score of 2	3	2	0	2	0	0
Score of 3	3	6	7	5	8	7

Table 15: INVEST Criteria Scoring Distribution for 8 User Stories, Case E

Initially, there was a problem where user stories were often difficult to estimate due to a lack of comprehensive elaborations and challenging to test because of unclear acceptance tests. However, this issue has now been mostly resolved, with five 3/3 scores for the Estimable criterion, with less user stories

than Workshop 1, and also seven 3/3 scores for Testable. As before, the user stories are generally considered to be negotiable, valuable, and well-sized.

Answering SRQ 5, the application of the INVEST criteria in this project demonstrated significant effects, leading to improvements for the Estimable and Testable criteria. The Small criterion consistently received the highest scores, albeit showing no improvements over previous user stories. On the other hand, we see no considerable improvements for the Negotiable criterion, even observing a slight decline regarding independence.

9.3 Reflection

In this case study, due to the absence of a developer overseeing both Workshops 1 and 3, we reflected on the experiment with only the onboarded PO. Despite this deviation from the research design, the PO found the experiment beneficial, valuing the opportunity to pause and reflect on their work process. They stressed the importance of being open to changes and ensuring the development team understands this flexibility. Areas for improvement were identified for Independent, Estimable, and Testable, with more workshops needed for the former. Testability was challenging due to differing perspectives between developers and the PO. While Scrum promotes small user stories, more detailed ones are needed for Testable. Despite these challenges, the PO appreciated the increased awareness and understanding from the experiment but noted the framework needs more focus on legal compliance. They would recommend using the criteria, especially to better guide business users.

In addressing SRQ 6, the PO noted a significant benefit is the increased awareness and understanding of INVEST among team members. However, challenges include the practical difficulties in ensuring user stories are truly independent, especially when dependencies exist between teams or applications. Testability remains a complex concept, as the developer and PO often have different approaches and understandings. Despite these challenges, the INVEST framework contributes to a more structured and thoughtful process for defining user stories. However, INVEST might require additional customization, such as legal compliance checks, to fit specific organizational needs.

10. Results

10.1 Results per case

In this chapter, we focus on the impact of the study's measurements by analyzing the user story scores for each criterion in the INVEST grid. Using stacked bar charts, we compare the distribution of scores on a 0-3 scale from Workshop 1 to Workshop 3, ensuring a clear and consistent assessment of the user story scores aligned with the INVEST criteria. This visualization allows us to identify patterns per case and assess whether improvements are universal or case-specific.

10.1.1 Case A

The workshops in Case A partially met expectations, shown in Table 16, with Testable improving notably, with no user story getting a 0/3 or 1/3 score. Yet it remained challenging to permanently change the way Testable is addressed due to its time-consuming nature. Independent and Estimable improved sufficiently to eliminate further workshops, although the former remained difficult to structurally change. Minimal changes in Negotiable and Valuable, and no change in Small were observed as these received high scores both prior to and after experimental intervention. Both Independent and Negotiable were deemed redundant and time-consuming by the team.

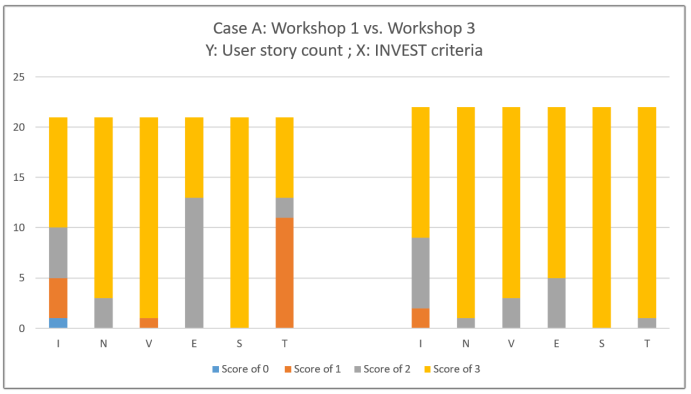

Table 16: Case A: User Story Counts by Score Level

10.1.2 Case B

In Case B, the workshops improved all criteria, with Independent, Small, and Testable achieving only 3/3 scores, as shown in Table 17. Previously, dependencies were an issue, but this concern was eliminated. However, there was a decline in Negotiable due to non-negotiable API-related user stories, with 3 out of 14 user stories not receiving a 3/3 score. Valuable, often skipped before, improved greatly post-experiment, with the developer praising the PO for enhancing the team's understanding. Although Estimable was difficult to assess and Small and Testable needed focused efforts, all criteria showed great flexibility in implementation, as seen from the improved scores.

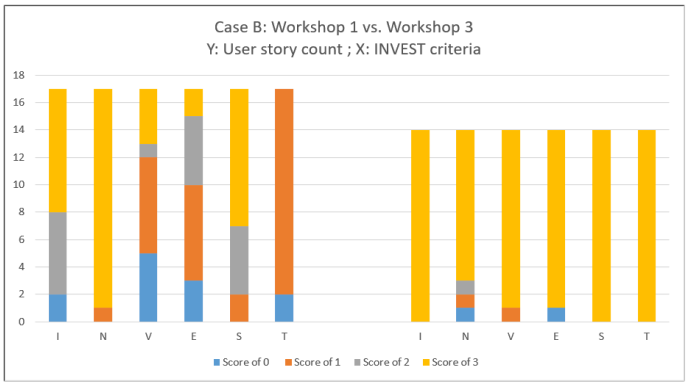

Table 17: Case B: User Story Counts by Score Level

10.1.3 Case C

The workshops aimed to create clearer, more independent user stories for Case C. We find slight improvements for Negotiable and Estimable, and major improvements for Valuable and Testable, as shown in Table 18. Changes in Independent and Small were considered too small, as managing dependencies remained a challenge. The PO valued negotiability and estimability but struggled with consistency. While Valuable was easier to address, Testable showed potential for many quick wins. Despite some challenges, we take from the table that the workshops improved the quality of user stories.

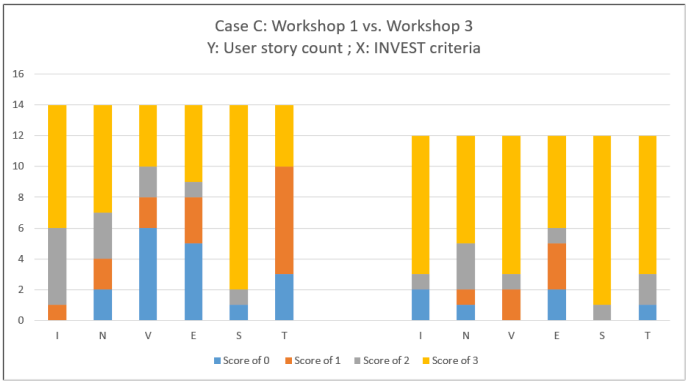

Table 18: Case C: User Story Counts by Score Level

10.1.4 Case D

While the experimental group was considerably smaller than the control group, we find that the workshops improved the scores for all criteria of Case D. Still, the PM viewed Independent and Negotiable as redundant. As shown in Table 19, Valuable achieved the 3/3 score for all user stories, which the PM emphasized as a crucial criterion. Estimable, Small, and Testable were considered straightforward criteria, with the observation that they are not exclusive to the INVEST framework.

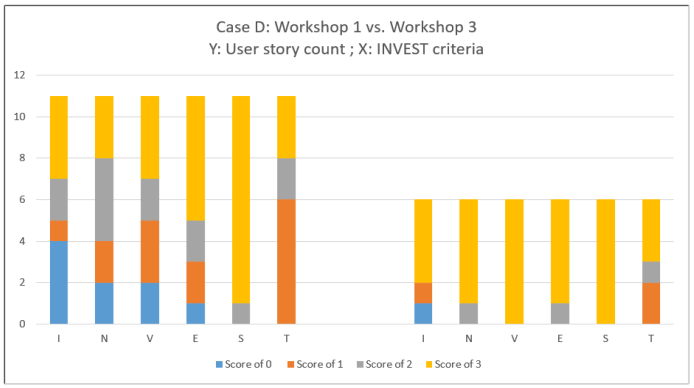

Table 19: Case D: User Story Counts by Score Level

10.1.5 Case E

We experienced mixed successes in Case E, as seen in Table 20, where Independent worsened after the experimental intervention. User stories from the control group had fewer dependencies compared to those from the experimental group. This difference is attributed to the varying nature of the user stories, as the ones of the experimental group are part of a new application with different data storage and retrieval systems. Despite these challenges, we find that both Estimable and Testable improved notably, indicating these criteria's flexibility to adjustments. Negotiable, Valuable, and Small already scored consistently high. In this case, the PO found it challenging to balance the level of detail required for testability with the Agile principle of (quote) *"working software over comprehensive documentation."* The amount of detail necessary to achieve high testability often contradicts this principle, creating a conflict in prioritizing between detailed user stories and maintaining focus on working software.

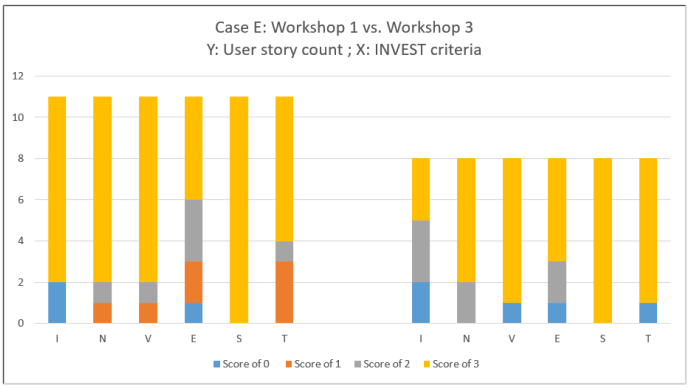

Table 20: Case E: User Story Counts by Score Level

10.2 Generalizability of Results

In this chapter, we go over the generalizability of the results from our case studies, specifically focusing on the insights from Table 21, which summarizes the findings from Chapter 10.1. The left half of the table presents the combined results from all five sessions of Workshop 1, covering 74 user stories. The right half shows the results from Workshops 2 and 3, totaling 62 user stories. By analyzing the outcomes and determining their generalizability across various contexts, we aim to determine the extent to which these findings can be applied to different Agile projects beyond the immediate scope of our study.

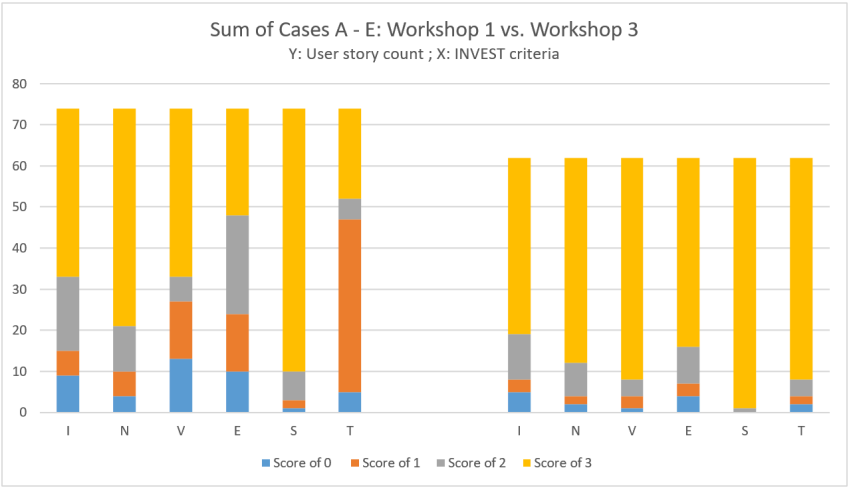

Table 21: Analysis of INVEST Criteria for Cases A-E Before and After Intervention

Table 21 shows that in Workshop 1, the scores across the INVEST criteria varied, with many user stories receiving lower scores (0 and 1) compared to higher scores (2 and 3), indicating inconsistency in quality. There were quite a few high scores (3), but the presence of lower scores suggested room for improvement. In contrast, Workshop 3 showed a more uniform and improved distribution, with most user stories receiving the highest score and fewer receiving lower scores. With this, we demonstrate a clear improvement in the quality and consistency of user stories, showing better adherence to the INVEST criteria and highlighting the progress made from Workshop 1 to Workshop 3 across Cases A to E.

Cases A, B, and D saw improvements for Independent, indicating success in reducing dependencies among those user stories. This suggests that focused strategies can help manage dependencies. However, in Case C, although more user stories received a 3/3 score after the experimental intervention, some user stories also received a 0/3 score during Workshop 3, unlike in Workshop 1 where no 0/3 scores were observed. Next to that, with Case E we experienced that the score distribution was negatively impacted, with the 0/3 score occurring as often as with Workshop 1, even though the experimental group contained less user stories. While Case E experienced this decrease, the PO foresaw this issue, as the user stories

from the experimental group were of a new application that is more dependent on other applications. In other words, the user stories were of a different nature with more dependencies between data storages and systems. These decreases for Independent highlight the challenges of maintaining independence in complex projects where dependencies with other teams or applications exist.

With considerable improvements regarding negotiability for Case C, where fewer scores of 0/3 and 1/3 were given, and Case D, where no scores of 0/3 or 1/3 were given, the experiment enhanced both flexibility and adaptability. However, Case B declined, indicating challenges in maintaining negotiability in rigid environments, in this case, where API-related user stories are picked up. Nonetheless, 11/14 user stories received a 3/3 score, showing high standards for this case. Cases A and E showed minimal changes, following the high scores found in Table 16 and Table 20. The results indicate that Negotiable depends on the project's flexibility and the team's adaptability, varying with each project's context.

For the Valuable criterion, Cases B, C, and D showed great improvements, visualizing that the experiment effectively enhanced the clarity of value delivered by user stories. This is in contrast to Cases A and E, which showed minor decreases, indicating that consistent focus is necessary to maintain a high standard. This may imply the need for additional support or training to sustain the enhancements for the other cases. Still, the two cases, A and E, scored considerably high before experimental intervention.

Regarding the Estimable criterion, all cases showed improvements, with Case B showing the biggest positive impact. This consistent enhancement suggests that providing detailed requirements and promoting clear acceptance criteria for user stories is universally beneficial to improve estimability.

Regarding maintaining a good user story size, Case B saw the most improvement, by receiving the maximum score for all user stories after experimental intervention, similar to Case D. Two teams, those of Cases A and E, had already received the maximum score for all user stories before the experiment. This suggests that the Small criterion is easily implemented with 61/62 user stories receiving a 3/3 score after experimental intervention.

The Testable criterion had the highest ratio of low to high scores among all the INVEST criteria for four of the five cases, indicating it needed the most attention during our experiment. While considered time-consuming, the Testable criterion showed consistent improvements across the entire study. We went from 42/74 user stories receiving a 1/3 score and 22/74 receiving a 3/3 score to 2/62 and 54/62 respectively. Case B saw the largest score increase, improving from 0/17 user stories with a 3/3 score to 14/14 for the Testable criterion. This was the largest improvement observed for any of the criteria across all five cases. This indicates that the experiment was very effective in making user stories more testable by defining clear acceptance criteria and test plans, where both happy and unhappy flows were considered.

11. Discussion

This discussion aims to map the observations from the individual case studies and then go over the findings noted across the research as a whole. We evaluate the case studies by exploring patterns and outcomes from each case, identifying challenges and adaptations, and assessing the impacts of the framework and the experiment. By distinguishing between the case-specific "Observations" and the more generalizable "Findings", we provide a clearer understanding of how the INVEST framework influences different project environments and its broader implications for ASD practices.

11.1 Case-Specific Observations

11.1.1 Case A

Obs. A.1 - User stories adhered better to INVEST but were not perceived as better by the developer.
Despite improvements in clarity, value, and testability, the practical outcomes did not reflect these enhancements. The developer found the extra context and acceptance criteria redundant, leading to duplication. This redundancy often resulted in extra text being skipped, especially for Testable. While the SC appreciated the extra context for Valuable, it did not prove helpful for developers in practice.

Obs. A.2 - Independent and Negotiable were seen as redundant and time-consuming.
The SC deemed the 'I' and 'N' of INVEST redundant. Independent was considered impractical as most user stories had dependencies. They found Negotiable added little value due to its complexity and the non-negotiable nature of some user stories. Therefore, the SC suggested focusing on Valuable, Estimable, Small, and Testable. This resulted in the SC proposing the VEST framework.

Obs. A.3 - The negotiability of a user story depends on the author.
The developer noted that the Negotiable score varied based on who wrote the user story, with the SC being praised for their flexibility and openness to feedback and suggestions.

Obs. A.4 - The Valuable criterion, post-experiment, was perceived differently within the team.
Although the developer did not notice a difference, the SC found that adding more context was beneficial for transparency and helped the team, and business users, better understand and refine user stories. This suggests that the impact of additional context might be more evident to those overseeing the process rather than to the developers themselves.

Obs. A.5 - The Estimable criterion remained challenging to improve on.
Most user stories still contained ambiguities that required follow-up questions to the SC. This lack of details often led to incorrect estimations, as the development team did not fully understand the user stories initially. The need for follow-up questions led to lower scores for Estimable.

Obs. A.6 - The Testable criterion was considered time-consuming and is therefore skipped.
The SC decided not to focus on the Testable criterion after experimental intervention due to the great amount of time it required during the experiment. This reflects a practical consideration, where they prioritize other criteria that they consider more manageable within their project's time constraints.

Obs. A.7 - Artificial Intelligence is applied to write user stories and to extend functionalities.
By applying AI, the SC found valuable assistance in writing user stories. The AI tool suggested acceptance criteria and refined them, making the user stories more complete and testable. This approach has been beneficial for business users involved in testing, as it adds context to the user stories and explains various ways a specific solution affects the application.

Obs. A.8 - Applying INVEST is recommended for future projects.
The SC mentioned they recommend INVEST for future projects, and also to colleagues. They strongly emphasized developing and implementing methods to better integrate INVEST into daily work routines. This includes creating structured guidelines, providing training sessions for team members, and integrating the criteria into regular project management and development processes.

11.1.2 Case B

Obs. B.1 - The user story quality had greatly improved post-experiment.
The developer highlighted the success of INVEST in enhancing the quality of user stories. Before the experiment, user stories were considered acceptable but too often lacked the depth and clarity required for efficient development. The team mentioned they had a much improved foundation to work with post-experiment. The PO was praised for always being open to learning and they themselves appreciated the structured reflection.

Obs. B.2 - Explicit value writing improved efficiency of both the sprint and refinement sessions.
Prior to the experiment, the PO often neglected the Connextra template, resulting in user stories with only acceptance criteria and no meaningful context, which led to superficial discussions. With INVEST, the PO provided both context and reasoning. This change made the developer's work more efficient, allowing them to better understand the user stories and ask more in-depth questions, resulting in better solutions. In one sprint, the team nearly completed all user stories with half of the sprint left, which the developer attributed to the improved user stories. Additionally, while refinements often ran over time due to lengthy discussions on specific user stories, the session following Workshop 2 finished with time to spare.

Obs. B.3 - The Independent criterion was not considered to be higher priority over other criteria.
The developer mentioned that while independence is important and is considered an issue within the project, it should not be prioritized due to issues with higher priority such as overly large user stories and insufficient testability. During the entry interview, it was revealed that some user stories were so large they had to be split into multiple smaller ones during refinement sessions. We experienced this as well, with one user story divided into seven smaller stories during the second workshop. By splitting these user stories, the dependencies became more manageable.

Obs. B.4 - Initial issues with the Testable criterion improved through basic testing guidelines.
Initially, the Testable criterion was not well implemented. The PO believed that detailed test plans should not be included by them since a dedicated tester was also part of the team. However, this approach overlooked the importance of having clear indications of how the PO would test and accept the user story. During the experiment, the PO incorporated alternative flows within the user stories, improving their scores for Testable to the maximum. This change ensured that even without a detailed test plan, the user stories contained enough information to guide the testing process, resulting in better quality assurance.

Obs. B.5 - Splitting large user stories made refinement sessions more valuable.
Before INVEST was applied, refinement sessions were often consumed by the task of breaking down large user stories and assigning them to developers, which took valuable time away from in-depth discussions. However, because of the experiment, user stories were already appropriately sized before the refinement sessions. This ensured discussions during the refinement were clearer, and less time was needed for clarifications, allowing the team to focus on content and deeper issues within the user stories.

Obs. B.6 - The INVEST checklist was considered too general and not usable.
The PO noted that the INVEST checklist was too general and not usable, as its guidelines lacked depth. The specificity needed to address the unique challenges and requirements of their project was not considered in the checklist. This highlights the necessity of customizing such frameworks to better fit specific project contexts and developing more detailed, actionable guidelines.

Obs. B.7 - INVEST is endorsed for future projects.
The PO mentioned they would endorse INVEST for future projects or upcoming POs, highlighting the positive impact on the team's understanding of user stories. While they did not notice changes in the team's working efficiency, they appreciated the study and also improved their understanding of the needs of the readers of their user stories.

11.1.3 Case C

Obs. C.1 - The team faced challenges with dependencies due to their project complexity.
The project of Case C was characterized by complex dependencies. These dependencies were challenging to manage and often could not be eliminated, requiring the team to find ways to work around them instead. This complexity impacted the team's ability to achieve truly independent user stories, as many functionalities were considered interconnected.

Obs. C.2 - Improvements for Valuable enhanced team understanding and engagement.
User stories previously lacked context, causing developers to follow instructions blindly. The PO stressed adding context, and the developer appreciated this change. After the experiment, user stories included detailed value explanations, helping the developers to understand task rationale. This enhanced discussions and critical thinking during development, as the team better understood the user stories' purposes.

Obs. C.3 - The Negotiable criterion is complex and perceived differently by the team.
Assessing negotiability was challenging due to its subtle differences within the INVEST grid, observed by the developer. The developer noted that the criterion, as described, lacked clarity because there is always some degree of negotiability, especially with their flexible PO. Although user stories typically scored well on this criterion, the reality was more complex. Different user stories required varying levels of detail and granularity, making it difficult to apply a one-size-fits-all approach. This observation highlighted the need for a more refined and context-sensitive evaluation method for Negotiable.

Obs. C.4 - Inconsistent level of detail for user stories made scoring for Estimable challenging.
The scores for Estimable remained difficult to apply consistently, largely due to the varying levels of detail in user stories. The developer mentioned they frequently underestimated due to ambiguities in the user stories in the past. However, the workshops helped the team recognize the importance of identifying and addressing these ambiguities early, particularly during refinement meetings. As a result, refinement sessions became more valuable, as the team learned to search for and address these points beforehand, leading to more accurate estimations and better-prepared development processes.

Obs. C.5 - Improving user story testability allowed for quick wins.
The area of Testable was considered an area where many "quick wins" could be achieved. Simple adjustments, such as adding specific acceptance criteria and expanding on what is already given, greatly improved testability.

Obs. C.6 - INVEST will be implemented across other project teams.
The PO expressed a commitment to implementing the INVEST framework across other teams within the company. After taking over for another PO on leave and finding the quality of their user stories lacking, the PO planned to conduct a similar training and implementation process upon their return. This initiative aimed to enhance the quality of user stories by focusing on criteria like Valuable and Estimable, ensuring that all teams benefit from the improved clarity and context provided by the INVEST framework.

11.1.4 Case D

Obs. D.1 - User stories adhered better to all criteria of INVEST post-experiment.
With the PM, we observed that the user stories adhered better to INVEST after Workshops 2 and 3. Although the PM's writing style did not change drastically, the workshops showed the adaptability of user stories and the flexibility of the INVEST criteria.

Obs. D.2 - Incomplete testing information made it difficult to enhance the testability.
Testability remained a challenge because user stories often lacked either explicit or complete details on how they would be tested. The PM observed that while the acceptance tests might not be entirely covered, they could not immediately identify additional testing flows. This highlighted the need for more structured guidance on enhancing the testability of user stories.

Obs. D.3 - Independent should focus on managing dependencies rather than eliminating them.
For the Independent criterion, the PM emphasized the importance of identifying and managing dependencies rather than eliminating them. In a sector as complex as insurance, dependencies are considered unavoidable, according to the PM. The focus should be on clearly defining these dependencies to manage them effectively rather than trying to make each user story completely independent.

Obs. D.4 - Independent and Negotiable often do not fit the scenarios as the INVEST grid suggests.
The PM felt that the theoretical explanations in the INVEST grid did not always align with real-world scenarios. Independent should focus on managing dependencies, and Negotiable should consider the scope and boundaries of the user story. This perspective shows the need for a more flexible and context-sensitive approach to applying these criteria. As a result, the PM found these criteria redundant.

Obs. D.5 - The Valuable criterion improved developer understanding and engagement.
Including the Valuable criterion more consistently into user stories had a positive impact. The PM noted that developers appreciated the added context, which helped them understand the purpose and value of their developments better. This led to more thoughtful development and potentially better solutions.

Obs. D.6 - INVEST is being viewed as a flexible tool rather than a set of strict rules.
The PM mentioned that the Independent and Negotiable criteria are redundant due to the theoretical nature, which did not always translate well to practical, real-world scenarios. Other criteria, namely Estimable, Small, and Testable, are universally applied to different Agile principles. With that, the PM took from the framework what they found useful and unique, which was the Valuable criterion.

Obs. D.7 - Emphasis should be on quantitative analysis for long-term benefits.
The PM suggested that future studies should include a greater emphasis on quantitative analysis to assess the long-term benefits of the INVEST framework. They proposed tracking metrics such as story points, hours spent per story point, and standard deviation in those values to evaluate improvements in estimation accuracy over time. While this approach requires significant resources, it could provide a more objective measurement of the framework's effectiveness in practice.

11.1.5 Case E

Obs. E.1 - Project and system variability impacted user story independence.
The team faced challenges with user story independence due to dependencies on other teams and applications. The second and third workshops' user stories had more dependencies than those discussed in the first workshop, due to being part of a newer project with other data storage and retrieval systems, complicating the dependencies. These dependencies fell outside the scope of the INVEST grid, and despite efforts to manage them, the team was unable to eliminate these.

Obs. E.2 - The PO suggested including a "Legal compliance" criterion.
Given the strict government regulations in the energy sector, legal compliance was a critical aspect for this project. Currently, legal compliance is managed by including it in the DoR. The team felt that including legal compliance more explicitly within the framework would better fit their needs. This adaptation would help ensure that legal compliance would be considered for all of their user stories.

Obs. E.3 - Considering alternative flows improved user story completeness.
By considering happy, unhappy, and alternative flows, we improved the work's completeness. This also involved prompting the PO to consider and discuss these additional scenarios. While the impact on workflow and project outcomes was not immediately noticeable, these discussions were deemed valuable.

Obs. E.4 - Balancing comprehensive documentation and testability is considered challenging.
The PO felt challenged in ensuring user stories had enough detail to be testable while still aligning with Agile's focus on working software rather than extensive documentation. This suggests a need for improved methods to balance these priorities.

Obs. E.5 - INVEST is recommended to better guide business users.
The PO would recommend using the INVEST criteria, especially to better guide business users. This follows from discussing the improvements made for the Testable criterion.

11.2 Generalized Findings

Following the observations made per case, we summarized them into generalized findings. These findings discuss the improvements observed in user story quality, the challenges encountered during the implementation of the framework, the adaptations suggested for specific project needs, and the long-term impacts of these changes on the teams.

Finding 1: Post-experiment user stories have improved in quality.
Following Obs. B.1 and Obs. D.1, the application of the INVEST framework led to improved user story quality. This was especially noticeable for the Valuable, Estimable, and Testable criteria. While each case faced unique challenges, the enhancement in user story quality was a common outcome.

Finding 2: Tackling dependencies was considered challenging and often not prioritized.
The Independent criterion was challenging across several cases, as we extract from Obs. A.2, Obs. B.3, Obs. C.1, Obs. D.3, and Obs. E.1, with teams not prioritizing the elimination of these dependencies. Dependencies on other teams or systems made it difficult to create independent user stories. Instead, the focus shifted to managing and clearly defining these dependencies.

Finding 3: The Negotiable criterion was considered to be practically limited.
The Negotiable criterion was found to be less practical in certain cases, resulting from Obs. A.2, Obs. C.3, and Obs. D.4. The theoretical explanation in the INVEST grid did not always align with real-world scenarios, where user stories often needed to be flexible within their defined scope but were not always negotiable. For example, API-related user stories often depend on other teams, and therefore the contents cannot be negotiated by developers. For such requests, it quickly becomes a technical specification.

Finding 4: Consistently considering the user story value improved understanding and engagement.
Developer understanding and engagement improved after the value and context were added more frequently and consistently following Obs. A.4, Obs. B.2, and Obs. D.5. Developers appreciated this, as it helped them understand the purpose and importance of their tasks better, leading to more thoughtful development and improved solutions.

Finding 5: Clear acceptance criteria improve user story estimability.
We have found that user stories, where acceptance criteria were more extensively written down, were considered better estimable. We observed this through Obs. A.5 and Obs. C.4. By better understanding the scope of user stories, more accurate estimates are made.

Finding 6: User story size is effectively managed.
The experiment showed noteworthy improvements with Obs. B.5 and Obs. D.1 in managing user story size. Most teams effectively managed story size before the experiment, with further improvements observed afterward. This suggests that the Small criterion is easily implemented.

Finding 7: The Testable criterion remained challenging and time-consuming.
Testability was a recurring challenge in several cases, namely Obs. A.6, Obs. D.2, and Obs. E.4. While improvements were noted, also through Obs. B.4 and Obs. C.5, Cases A, D, and E struggled with explicitly defining how user stories should be tested in a timely manner. This often led to inconsistencies and highlighted the need for more structured guidance on how to implement this into user stories.

Finding 8: The acronym was considered a toolkit to select from, not a strict set of rules.
Most teams mentioned that certain criteria are more relevant than others, leading them to focus on specific points. We saw this with Obs. A.2, Obs. D.6, and Obs. E.2. Adaptations, such as the proposed VEST framework in Case A and D, and the inclusion of legal compliance for Case E, were crucial for managing practical realities and unique project needs.

Finding 9: The experiment increased awareness of INVEST and structured user story writing.
The framework increased awareness and understanding among team members, which contributed to a more structured and thoughtful process for discussing user stories. We figured this out from Obs. B.1, Obs. B.2, Obs. C.2, Obs. D.5, and Obs. E.3. This structured approach helped create a common ground for quality standards and improved the workflow. More importantly, we observed improved refinement sessions.

Finding 10: Repetitive issues in user stories highlight the need for the adoption of INVEST.
For developers, Workshop 1 became frustrating as they encountered the same issues repeatedly with different user stories, following Obs. A.1 and Obs. D.4. This repetition underscores the potential benefits of successfully adopting the INVEST framework to improve the quality and consistency of user stories.

Finding 11: Active participant involvement enhanced the effects of the workshops.
Experiment activities went smoothly due to the openness of the POs and the engagement of the developers. Following Obs. A.3 and Obs. B.1, the active involvement of the POs and their responsiveness to feedback positively impacted the effectiveness of INVEST. This highlights the importance of continuous feedback loops between POs, developers, and the researchers.

Finding 12: INVEST is recommended for Agile software development project teams.
The empirical evidence from the study strongly supports the use of the INVEST framework in Agile software development projects. Following Obs. A.8, Obs. B.7, Obs. C.6, and Obs. E.5, the INVEST framework is recommended for Agile teams seeking to optimize their user story formulation and project efficiency.

12. Conclusion

Our study examined how the INVEST framework can improve user story quality and project outcomes in Agile software development. At first, we focused on identifying effective methods to measure the impact of the INVEST framework at both the user story and project levels. We conducted a literature review and analyzed metrics such as burndown charts, velocity tracking, exit surveys, and interviews. Metrics like Function Points and Story Points provided objective measures of size and complexity. These findings highlight the importance of combining quantitative and qualitative measures for a thorough evaluation of user story success. Through this approach, we were able to examine what the most effective tools are for assessing the practical usefulness of user stories in SRQ 1.

Then we explored the practical flexibility and applicability of INVEST across different projects, through SRQ 2. We reviewed previous studies where they had conducted workshops with Agile teams to see how those teams adapted the criteria to their unique needs, highlighting the adaptability of INVEST. Criteria like Small and Testable were emphasized, while Independent and Negotiable were less frequently considered. These adaptations provided insights into how teams implement Agile methodologies based on their unique project needs, while also corresponding with the findings of our embedded experimental-case study design. This showed that while some aspects of INVEST are universally beneficial, others need to be tailored to specific project contexts. This result has been identified while answering both SRQ 2 and SRQ 6, showing that our findings from literature complement our findings from our embedded experimental-case study design.

Following this up with SRQ 3, we explored practical lessons and takeaways from empirical studies to provide actionable strategies for Agile teams. We analyzed empirical data of case studies from literature along with workshops they facilitated to gather insights. With this, we aimed to make the theoretical principles of INVEST more applicable and effective in practice. Successful implementations involved strong communication, regular meetings, and balancing researcher goals with team dynamics and organizational culture. These findings served as the basis for our research design. This demonstrated that consistent application of INVEST, supported by clear communication and regular review, leads to improvements in the research process, addressing SRQ 3.

Following up on the first three SRQs, we continued with our embedded experimental-case study based on these findings. Studying SRQ 4, we established a baseline understanding of user stories to identify common challenges and areas for improvement. We considered this crucial for measuring the framework's impact later on and providing tailored support to the Agile teams. The teams under study often struggled with creating independent, valuable, estimable, and testable user stories. Workshops revealed common issues, such as difficulties maintaining independence and ambiguity, which hindered development. These findings highlight where Agile teams need the most support in implementing the INVEST framework effectively. This initial phase was vital in identifying the potential improvements.

To understand the practical flexibility of the INVEST framework, SRQ 5 revealed that while the INVEST criteria are useful, strict adherence is not always practical. We monitored the application of INVEST in various projects and gathered feedback from the teams. We found that Valuable, Testable, and Estimable are generally adaptable and showed improvement when applied. However, the Independent and Negotiable criteria proved less adaptable due to dependencies and project constraints. This indicated that while the principles of INVEST are sound, their application needs to be flexible to account for real-world complexities, as explored in SRQ 5.

Following the experimental intervention, we addressed SRQ 6. We collected feedback through reflections with our research participants to understand the challenges and benefits of implementing INVEST. Key challenges included the initial learning curve for specific INVEST criteria, such as Independent and Testable, and the occasional difficulty in balancing all criteria at once. Overcoming these obstacles required continuous feedback from the team. Despite these challenges, the framework's benefits were

evident: clearer user stories, and both improved refinement sessions and team communication. We found that recognizing these challenges helps anticipate and address potential issues when adopting INVEST.
The six SRQs played a role in collectively addressing the MRQ. Each SRQ was designed to explore a different dimension of the INVEST framework's application and its impact on the Agile software development project teams. Our mixed-method approach, combining literature review, workshops, and reflections, provided an extensive understanding. The MRQ itself was stated as follows:

> **Main Research Question:** *"How does the INVEST framework impact the practical usefulness of user stories in real-world Agile software development projects?"*

To conclude our study, we have found that the INVEST framework greatly enhances the impact the usefulness of user stories in real-world Agile software development projects. The framework primarily improved the clarity, value, and testability of user stories, leading to more effective refinement sessions and enhanced team collaboration. Although initial adoption presented challenges, such as the time and effort needed to ensure all criteria are met and the learning curve associated with understanding and applying the framework, these are outweighed by the benefits. By applying INVEST, Agile teams can achieve better alignment with user needs, more accurate estimations, and ultimately, higher-quality software development outcomes. Our approach highlighted that the INVEST framework is not just a theoretical tool but a practical solution that improves the quality and effectiveness of user stories when properly implemented and adapted to specific project needs.

13. Limitations and Future Work

Within this concluding chapter of our research, we provide an overview of the constraints and potential biases that may have influenced the study's results. This section aims to enhance transparency, address the reliability and validity of the findings, and help readers contextualize the results. In addressing these limitations, we make use of the framework provided by Creswell and Plano Clark (2017). Their approach categorizes validity into internal validity and external validity, providing a structured way to analyze and present the limitations of our study.

Internal validity refers to how confidently we can say that the results of the study are due to the interventions tested and not other factors. It involves ensuring that the observed outcomes are directly caused by the variables we manipulated and not by external influences. External validity, on the other hand, concerns the generalizability of the study's findings. It addresses whether the results observed in the study sample can be applied to a broader context. Threats to internal validity can include biases in data collection, participant selection, and data interpretation, while those of external validity include the characteristics of the sample, the context in which our study is conducted, and the time period during which the study takes place (Creswell & Plano Clark, 2017).

13.1 Internal Validity

Within this chapter we discuss challenges one would encounter when attempting to replicate this study, and which factors may differ when doing so.

Internal Limitation 1: We did not focus on quantitative data because of a lack of data being stored.
While we foresaw the analysis of metric data, as extensively discussed in Chapter 4.2.3, the participating projects have shown us that this is not always possible. Certain data is not being logged in the project management tools, such as issues not being linked to user stories or hours spent on a certain user story not written down. This made it, for example, too complex for researchers to find connections or correlations between the number and complexity of issues logged before and after the experimental intervention. For this reason, we did not rely as heavily on quantitative data as expected pre-experiment.

Internal Limitation 2: The involvement of researchers could vary per study.
Other researchers might not have gone as in-depth as we did during the workshops. We were thorough in discussing different flows of the described functionality to ensure user story completeness. The follow-up questions were user story specific, and could have resulted in different scores had it been other researchers taking care of those workshops. This could also be affected by the time-investment from the project teams' perspective.

Internal Limitation 3: Developer reassignment introduced variability.
Initially, for every case, we onboarded a specific developer. However, due to differing circumstances, such as reassignment to different projects and vacations, we had to onboard a new developer from the same project team for Cases D and E to ensure continuity of our research activities. Although this transition introduced a variable that could potentially impact the results, we were able to continue the research without large delays. Nonetheless, we observed that the scores assigned using the INVEST grid varied greatly between Workshop 1 and Workshop 3. It is unclear whether this difference resulted from the improvements or differing interpretations by developers.

Internal Limitation 4: User stories written by developers lacked detail.
When reviewing user stories for Case E, we found that some user stories were written by developers themselves to speed up the process. The PO presented these user stories, which lacked many details, yet developers found them to suffice. Even though these user stories were not included in the experiment itself, this raises a question: why do these detailed-lacking user stories suffice for developers, but those written by the PO do not? This inconsistency highlights the challenge of going with the subjective opinions of developers regarding the sufficiency of user stories.

Internal Limitation 5: Incomplete assessment of user story independence.
When considering independence, we only assessed whether a user story could be developed and released independently of other user stories. We did not consider whether the functionality affected other functionalities in the application, which could increase the workload for regression testing activities. This oversight means that even if user stories appear independent on the surface, their implementation might still require significant integration and regression efforts, potentially nullifying the benefits of having independent user stories.

Internal Limitation 6: Ambiguity in defining dependencies.
Considering independence, almost every user story depends on some previously built functionality. However, it is unclear when a previous functionality should still be considered a dependency - whether it was built months ago, five sprints ago, or last sprint. With no clear-cut answer, this ambiguity complicates the process of identifying and managing dependencies that need to be addressed first.

Internal Limitation 7: Overly small user stories not considered.
User stories can sometimes be too small, and this aspect is not addressed within the INVEST grid. While the INVEST criteria emphasize the importance of keeping user stories small and manageable, there is a risk of excessive splitting. This can result in numerous tiny user stories that complicate the development process, increase the overhead in tracking and managing tasks, and potentially reduce overall efficiency and coherence in the project.

Internal Limitation 8: Efficiency metrics may not account for all factors.
For Case B, we observed improved efficiency, with more user stories completed in fewer days. However, it could have been that the developers spent more hours per day on the project, according to the PO. This additional effort is not accounted for in the metrics used, making it difficult to attribute the observed efficiency improvements solely to the application of the INVEST criteria.

Internal Limitation 9: Velocity and burndown metrics are not always appropriate for analyzing progress.
In hindsight, velocity or burndown metrics are not always appropriate for analyzing progress, as sprint velocity could depend on more than just the difficulty or understandability of the user stories. Factors such as employees taking days off for holidays or illness can also affect these metrics, or even the opposite, where developers allocate more hours to developments. This makes these metrics unreliable indicators of progress related to user story quality.

Internal Limitation 10: Reliance on qualitative data introduces bias.
The reliance on qualitative data from interviews and workshops may affect the objectivity of the findings, as interpretations of the data can vary based on individual perspectives and experiences.

Internal Limitation 11: Potential bias in self-reported data.
The study relies on self-reported data from developers and POs, which can introduce bias. Participants may overestimate the positive impacts or underreport the challenges they faced, affecting the accuracy of the findings. Additionally, they selected the user stories themselves. Although they claimed the selection was random, we are unsure to what extent this randomness was maintained.

Internal Limitation 12: Lack of enforced changes.
We did not enforce changes to the user stories during the study. Whenever we identified an issue and put a cross during Workshop 2, it did not automatically result in improvements. This limitation highlights that merely identifying issues does not guarantee that they will be addressed, and active enforcement or follow-up may be necessary to ensure improvements are made. As a result, oftentimes criteria like Independent or Testable, which required more initiative from the POs perspective, were left unaddressed.

Internal Limitation 13: Bias due to researcher expertise.
The fact that the main researcher has a background in functional testing could be considered a limitation. This expertise might have led us to ask more follow-up questions, potentially inflating the Testable scores. Our familiarity with testing could unconsciously have influenced the questions asked, focusing more on areas we deem important rather than maintaining an objective stance.

Internal Limitation 14: Imbalance between low-code and full-stack teams.
Four out of the five participating teams used low-code platforms, while the one full-stack team, that of Case C, faced complex dependencies. This difference in technologies and project complexity might affect the study's findings, as low-code and full-stack projects have distinct development processes and challenges. This predominance of low-code teams may potentially affect the consistency of the results.

Internal Limitation 15: Subjectivity in user story evaluation.
Evaluating user stories involves a degree of subjectivity, as different developers might interpret the criteria, the scores, or even the descriptions of user stories differently. This subjectivity can introduce variability in the scoring and assessment process. In Cases D and E, different developers participated in Workshops 1 and 3, resulting in great score improvements. It is uncertain whether these improvements were due to the experiment or the differing interpretations of the developers.

Internal Limitation 16: Potential learning curve effects.
The familiarity of participants with the INVEST criteria and the workshop process could have improved over time, leading to potential learning curve effects. As participants became more familiar with the process, their evaluations and interactions might have changed, affecting the consistency of the results across different time points.

Internal Limitation 17: Limited data sources for literature review
For our data collection we leaned towards the Google Scholar search engine for the literature review. This may have resulted in a narrower scope of reviewed literature, potentially overlooking relevant studies published in other academic databases or sources.

Internal Limitation 18: Variability in participants' experience levels could affect results.
The participants in our study had varying levels of experience with Agile methodologies and the INVEST framework. This variability could influence the scores of the user stories that were assigned, but also the degree to which user stories were adjusted during the second workshops., More experienced participants may adapt more quickly and effectively than those with less experience. Therefore, the results might not accurately reflect the impact of the INVEST framework across different experience levels within ASD teams.

13.2 External Validity

External validity, on the other hand, concerns the generalizability of the study's findings to other settings, populations, or times. It addresses whether the results observed in the study sample can be applied to a broader context.

External Limitation 1: Small user story sample size affects generalizability.
Case D contained only six user stories in the experimental group (Workshop 2, Workshop 3), making it difficult to generalize the results. Similar to the eight user stories of Case E, this might not have been sufficient to represent broader trends or outcomes effectively within the respective projects. This limited sample size challenges the reliability of the findings.

External Limitation 2: Convenience sampling during case study recruitment introduces bias.
We applied convenience sampling for recruiting case studies. This method, while efficient, may introduce bias because it does not ensure a random sample from the target population, that is, ASD projects in the Netherlands. Consequently, the findings might not be fully representative of all Agile projects or teams,

limiting the generalizability of the results. As a result, we only considered projects that are open to change, and given that the openness of the participants played a crucial role in applying this framework, the outcomes might have been different in less flexible or more resistant environments.

External Limitation 3: Participation declines due to limited time investment in preparing user stories.
During the process of recruiting projects to participate in the study, numerous potential cases declined due to their limited time investment in preparing user stories, typically dedicating only a single day before refinement to write these. While we found that these cases, in particular, would have benefited the most from participating, they did not see the added value of the framework, indicating their current projects are sufficiently efficient. Besides that, some projects declined because the experiment would cost them too much time which they could not allocate, even when the potential benefits would outweigh those costs in the long run. They preferred to allocate those hours to more immediate development tasks.

External Limitation 4: Limited study duration affected the long-term impact assessment.
Timeline constraints, primarily from the participants' project management schedules, limited the observation period of the study. This constraint made it difficult to assess the long-term impacts of the changes introduced during the study, thus providing only a short-term view of its effectiveness.

External Limitation 5: Lack of diversity in project types and industries.
With the study focusing on five projects, each from a different sector, the findings may not be fully generalizable to other project types or industries not represented in the study. While we do provide a diverse sample with five cases, it may not cover the full spectrum of project types and industries.

External Limitation 6: Limited generalizability due to differences in technologies.
Four of the five participating teams used low-code platforms for their developments. That is opposed to the full-stack team in Case C, who encountered complex dependencies. Most of our findings may not be fully applicable to other project types, especially full-stack projects. The complex dependencies in the full-stack team might be typical for such projects but are not represented in the low-code teams, limiting the broader applicability of our results.

External Limitation 7: Geographical focus on Dutch ASD project teams limits generalizability.
We only discussed Dutch ASD project teams in our study. While this focus provides detailed insights into the Dutch context, it limits the generalizability of our findings to other countries or even continents. Different cultural, economic, and organizational factors in other regions could influence the results the INVEST framework has on project teams.

13.3 Considerations for Future Work

Having discussed the limitations of the study, we also make several suggestions for future research to explore areas that could enhance the understanding and application of the INVEST framework. These suggestions also aim to provide guidelines for future studies to build on our findings and improve the practical implementation of similar frameworks in ASD projects.

Future Work 1: Investigate the role of Artificial Intelligence in writing user stories.
We did not explore how AI could assist with writing acceptance criteria or entire user stories in this study. However, we noticed that the SC of Case A used AI for this purpose, which they continued to do during Workshop 2. Investigating the potential benefits and drawbacks of AI in this context could provide valuable insights for future research and practice.

Future Work 2: Investigate specific contexts by narrowing the scope to certain industries.
To enhance the generalizability of the findings, for certain industries, future research should include more studies within the same industry. While this study focused on five projects from five different sectors, focusing on additional studies within the same industry will provide a deeper understanding of how the

INVEST criteria perform in specific contexts. Considering we have found that specific industries, such as that of Case E, identified shortcomings in the framework, we might find that other projects tend to agree.

Future Work 3: Enhance the assessment of user story dependencies.
Future research should focus on refining the methods used to identify and manage dependencies among user stories. This includes developing clearer guidelines on when a previously built functionality should still be considered a dependency, to reduce ambiguity and improve project planning. Additionally, exploring tools or techniques to better track and visualize these dependencies can support in understanding their impact on the workflow. Addressing these dependencies more effectively will also help in tackling risks related to integration issues.

Future Work 4: Enhance the assessment of user story negotiability.
Not all user stories are considered negotiable, which can limit their flexibility during development. Studies should focus on improving methods to assess the negotiability of user stories. Our findings suggest that negotiation is acceptable as long as it remains within the user story's scope. Clearer guidelines and tools will help teams adapt to changes while maintaining their scope.

Future Work 5: Evaluate long-term impacts of the INVEST framework.
Future studies assessing the long-term impacts will help in understanding how the benefits of the framework evolve over time. Long-term studies, including quantitative metrics, can also identify any potential drawbacks or areas needing adjustment to maintain effectiveness.

Future Work 6: Assess the impact of developer experience on the effectiveness of INVEST.
It is important to investigate how the experience levels of developers impact the application and outcomes of INVEST. This includes exploring whether more experienced developers benefit more from the framework compared to less experienced ones, and how training programs can address the needs of developers at different stages of their careers. Understanding this dynamic will help in optimizing the implementation and training processes.

Future Work 7: Improve metrics for evaluating user story quality.
There is a need to develop and refine metrics that more accurately evaluate the quality of user stories. Current metrics like velocity and burndown are not always reliable indicators of progress related to user story quality. Future studies should focus on creating metrics that better capture aspects such as user story completeness and testability to provide a more accurate assessment.

Future Work 8: Explore how to balance conflicts between INVEST criteria and Agile principles.
Exploring strategies to maintain a balance between conflicting INVEST criteria with Agile principles would improve the application of frameworks like INVEST. For instance, in Case E, ensuring detailed testability in user stories conflicted with the principle of favoring working software over documentation. Similarly, we found that when user stories decrease in size, their dependencies add up. There is a need to further investigate other potential conflicts with the INVEST criteria and find balanced solutions.

Future Work 9: Address the limited generalizability due to differences in technologies.
Future research should aim to include a more balanced representation of different technological approaches, such as full-stack projects. By expanding the scope to a wider variety of technologies and project types, future studies can provide a more rigorous understanding of how different technological contexts influence project outcomes. This would enhance the generalizability of findings and offer more robust insights applicable across development environments.

Future Work 10: Explore the VEST framework.
Investigating the VEST framework, and comparing VEST to INVEST in practical applications could provide valuable insights. Studies should examine VEST's integration into ASD. This will help determine if VEST offers advantages over INVEST and can better enhance project management.

References

1. Adali, O. E. (2017). Assess agility: agility assessment approach supported with an automated web based agility assessment tool (Master's thesis, Middle East Technical University).
2. Aitken, A., & Ilango, V. (2013, January). A comparative analysis of traditional software engineering and agile software development. In 2013 46th Hawaii International Conference on System Sciences (pp. 4751-4760). IEEE.
3. Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11).
4. Altameem, E. A. (2015). Impact of agile methodology on software development. Computer and Information Science, 8(2), 9.
5. Ananjeva, A., Paasivaara, M., & Behm, B. (2020a). User stories in agile software development. Journal of Systems and Software, 170, 110717.
6. Ananjeva, A., Persson, J. S., & Bruun, A. (2020b). Integrating UX work with agile development through user stories: An action research study in a small software company. Journal of Systems and Software, 170, 110785.
7. Anderson, D. J. (2010). Kanban: successful evolutionary change for your technology business. Blue Hole Press.
8. Anitha, P. C., Savio, D., & Mani, V. S. (2013, July). Managing requirements volatility while “Scrumming” within the V-Model. In 2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE) (pp. 17-23). IEEE.
9. Baham, C. (2016). The Impact of Organizational Culture and Structure on the Routinization of Agile Software Development Methodologies. https://dblp.org/rec/conf/amcis/Baham16
10. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance. https://agilemanifesto.org/
11. Borhan, N. H., Zulzalil, H., & Ali, N. M. (2022, November). A Hybrid Prioritization Approach by integrating non-Functional and Functional User Stories in Agile-Scrum Software Development (i-USPA): A preliminary study. In 2022 IEEE International Conference on Computing (ICOCO) (pp. 276-282). IEEE. DOI: 10.1109/ICOCO56118.2022.10031863
12. Buglione, L., & Abran, A. (2013, October). Improving the user story agile technique using the invest criteria. In 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (pp. 49-53). IEEE. Retrieved from https://ieeexplore.ieee.org/document/6693222
13. Cohn, M. (2004). User Stories Applied: For Agile Software Development. Addison-Wesley Professional.
14. Cowperthwaite, F., Horkoff, J., & Kopczynska, S. (2023). The Effects of Native Language on Requirements Quality. In Proceedings of the 18th Conference on Computer Science and Intelligence Systems (pp. 913–917). ACSIS, Vol. 35. Retrieved from: https://doi.org/10.15439/2023F9537
15. Creswell, J. W., & Clark, V. L. P. (2017). Designing and conducting mixed methods research. Sage publications.
16. Dalpiaz, F., & Brinkkemper, S. (2021, September). Agile requirements engineering: From user stories to software architectures. In 2021 IEEE 29th International Requirements Engineering Conference (RE) (pp. 504-505). IEEE. Retrieved from https://ieeexplore-ieee-org.proxy.library.uu.nl/abstract/document/9604656
17. Davis, C. W. (2015). Agile metrics in action: Measuring and enhancing the performance of agile teams. Manning Publications Co..
18. Dellsén, E., Westgårdh, K., & Horkoff, J. (2022, March). Invest in Splitting: User Story Splitting Within the Software Industry. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 115-130). Cham: Springer International Publishing.
19. Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. Journal of systems and software, 85(6), 1213-1221. https://doi.org/10.1016/j.jss.2012.02.033
20. Do Nascimento, S. S., Abe, J. M., Forçan, L. R., de Oliveira, C. C., Nakamatsu, K., & Ari, A. (2022, July). Improving the Process of Evaluating User Stories Using the Paraconsistent Annotated Evidential Logic Eτ. In International Workshop on New Approaches for Multidimensional Signal Processing (pp. 133-142). Singapore: Springer Nature Singapore.
21. Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. Academy of management journal, 50(1), 25-32.
22. Ernst, N., Bellomo, S., Nord, R. L., & Ozkaya, I. (2015). Enabling incremental iterative development at scale: Quality attribute refinement and allocation in practice. Software Engineering Institute, Tech. Rep. CMU/SEI-2015-TR-008.
23. Ferreira, A. M., da Silva, A. R., & Paiva, A. C. (2022, April). Towards the Art of Writing Agile Requirements with User Stories, Acceptance Criteria, and Related Constructs. In ENASE (pp. 477-484).
24. Fuksmane, N. (2023, January 12). Tracking scope changes and unplanned work with Insights for Jira Software Cloud. Atlassian Community. https://community.atlassian.com/t5/Jira-Software-articles/Tracking-scope-changes-and-unplanned-work-with-Insights-for-Jira/ba-p/2237140
25. Govil, N., & Singh, R. (2022). Agile methodologies in software development: A survey. Journal of Software Engineering and Applications, 15(3), 77-90.

26. Granda Juca, M. F., Alba Sarango, B. A., & Parra Gonzalez, L. O. (2021). Towards a model-driven testing framework for GUI test cases generation from user stories. In Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE (pp. 453-460). SCITEPRESS - Science and Technology Publications. Retrieved from: https://doi.org/10.5220/0010499004530460
27. Halme, E., Vakkuri, V., Kultanen, J., Jantunen, M., Kemell, K. K., Rousi, R., & Abrahamsson, P. (2021, June). How to write ethical user stories? impacts of the ECCOLA method. In International Conference on Agile Software Development (pp. 36-52). Cham: Springer International Publishing.
28. Highsmith, J. (2009). Agile Project Management: Creating Innovative Products. Addison-Wesley Professional.
29. Huijgens, H., & Solingen, R. V. (2014, June). A replicated study on correlating agile team velocity measured in function and story points. In Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics (pp. 30-36).
30. International Organization for Standardization. (2007). Information technology - Software measurement - Functional size measurement (ISO/IEC Standard No. 14143-1:2007). https://www.iso.org/standard/38931.html
31. Jeffries, R., Anderson, A., & Hendrickson, C. (2000). Extreme Programming Installed. Addison-Wesley Professional.
32. Jurisch, M., Lusky, M., Igler, B., & Böhm, S. (2017). Evaluating a recommendation system for user stories in mobile enterprise application development. International Journal on Advances in Intelligent Systems Volume 10, Number 1 & 2, 2017.
33. Kamath, D. (2023). Improving Agile Development Practices.
34. Khanh, N. T., Daengdej, J., & Arifin, H. H. (2017, February). Human stories: A new written technique in agile software requirements. In Proceedings of the 6th International Conference on Software and Computer Applications (pp. 15-22).
35. Kuhail, M. A., & Lauesen, S. (2022). User Story Quality in Practice: A Case Study. Software, 1(3), 223-243.
36. Larman, C., & Vodde, B. (2008). Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum. Addison-Wesley Professional.
37. Lin, G., Yang, M., Shao, Q., & Ma, L. (2014). Agile software development: A survey. Journal of Software Engineering and Applications, 7(10), 837-846.
38. Lucassen, G., Dalpiaz, F., Van Der Werf, J. M. E., & Brinkkemper, S. (2015, August). Forging high-quality user stories: towards a discipline for agile requirements. In 2015 IEEE 23rd international requirements engineering conference (RE) (pp. 126-135). IEEE.
39. Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2017). Improving user story practice with the Grimm Method: A multiple case study in the software industry. In Requirements Engineering: Foundation for Software Quality: 23rd International Working Conference, REFSQ 2017, Essen, Germany, February 27–March 2, 2017, Proceedings 23 (pp. 235-252). Springer International Publishing.
40. Martakis, A., & Daneva, M. (2013, May). Handling requirements dependencies in agile projects: A focus group with agile software development practitioners. In IEEE 7th International Conference on Research Challenges in Information Science (RCIS) (pp. 1-11). IEEE.
41. Neumann, M., Bogdanov, Y., Lier, M., & Baumann, L. (2021). The Sars-Cov-2 pandemic and agile methodologies in software development: a multiple case study in Germany. In Lean and Agile Software Development: 5th International Conference, LASD 2021, Virtual Event, January 23, 2021, Proceedings 5 (pp. 40-58). Springer International Publishing.
42. Nisyak, A. K., Rizkiyah, K., & Raharjo, T. (2020, October). Human related challenges in agile software development of government outsourcing project. In 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI) (pp. 222-229). IEEE.
43. Noel, R., Riquelme, F., Mac Lean, R., Merino, E., Cechinel, C., Barcelos, T. S., ... & Munoz, R. (2018). Exploring collaborative writing of user stories with multimodal learning analytics: A case study on a software engineering course. IEEE Access, 6, 67783-67798.
44. Patton, J. (2014). User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media.
45. Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. Procedia-Social and Behavioral Sciences, 175, 455-463.
46. Pokharel, P., & Vaidya, P. (2020, October). A Study of User Story in Practice. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (pp. 1-5). IEEE.
47. Poth, A., Kottke, M., & Riel, A. (2019, September). Scaling agile on large enterprise level–systematic bundling and application of state of the art approaches for lasting agile transitions. In 2019 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 851-860). IEEE.
48. Raharjana, I. K., Siahaan, D., & Fatichah, C. (2021). User stories and natural language processing: A systematic literature review. IEEE access, 9, 53811-53826.
49. Rehkopf, M. (n.d.). User Stories. Atlassian. Retrieved December 7, 2023, from https://www.atlassian.com/agile/project-management/user-stories
50. Rizkiyah, K., Nisyak, A. K., & Raharjo, T. (2020, September). Agile-Based Requirement Challenges of Government Outsourcing Project: A Case Study. In 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE) (pp. 267-273). IEEE.
51. Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.
52. Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org. https://www.scrumguides.org/scrum-guide.html
53. Sutherland, J., & Schwaber, K. (2013). The Scrum Papers: Nuts, Bolts, and Origins of an Agile Framework. Scrum, Inc.

54. Tona, C., Jiménez, S., Juárez-Ramírez, R., Pacheco López, R. G., Quezada, Á., & Guerra-García, C. (2022). Scrumlity: an agile framework based on quality of user stories. Programming and Computer Software, 48(8), 702-715. Retrieved from: https://link.springer.com/article/10.1134/S0361768822080199
55. Wake, W. C. (2003). Extreme Programming Explored. Addison-Wesley Professional.
56. Willamy, R., Nunes, J., Perkusich, M. B., Freire, A. S., Saraiva, R. M., Almeida, H. O., & Perkusich, A. (2016). A method to build Bayesian networks based on artifacts and metrics to assess agile projects. In SEKE (pp. 81-86).
57. Yin, R. K. (2009). Case study research: Design and methods (Vol. 5). sage.

Appendix A. Consent Form for Partaking in Experiment

Title of Research Project: INVEST in Success: A Deep Dive into User Story Optimization
Researcher: Ashot A. Grigorian **Research Institute:** Utrecht University

This form is for a study on the INVEST framework in Agile project teams. Thank you for sharing practical insights into Agile practices. Our study focuses on gathering Agile project management user stories and experiences. We will anonymize any sensitive information like client or company names to ensure confidentiality and privacy. Your participation includes interviews, observations, and Agile-related tasks. These activities are safe and consider your convenience. We plan to record these activities for transcription purposes, with transcripts stored separately in the study report for detailed analysis. The activities will take place either online or on-site, based on the interviewee's preference.

a. Entry Interview for both Product Owner and Developer (0.5 hours):
    i. Individual interviews with the Product Owner and Developer.
    ii. Topics: Background (in IT), project experience, familiarity with INVEST and similar methods.
b. Workshop for Developer (1.5 - 2 hours):
    i. We review completed user stories from their project's previous (and current) sprints.
    ii. We rate these user stories using the INVEST grid*.
    iii. We store these user stories and their grid ratings, labeling them "Control group".
c. Workshop for Product Owner (2 - 3 hours):
    i. The Product Owner brings in user stories of which we store the initial contents.
    ii. We apply the INVEST checklist* to enhance these user stories.
    iii. We store these enhanced user stories, labeling them "Experimental group".
d. Workshop after Refinement Meetings (0.75 hours):
    i. The team members discuss the user stories from the experimental group.
    ii. The goal is to have developers rate the experimental group's user stories using the INVEST grid.
e. Researcher Access:
    i. The Researcher gains access to the Sprint board for tracking progress.
    ii. The Researcher gains access to the user stories for tracking issues and discussions.
    iii. The Researcher is allowed to store metric data and related discussions.
f. Exit Interview (0.5 hours):
    i. Individual interviews with the Product Owner and Developer.
    ii. We focus on the experiences and feedback of the involved team members.

*This will be provided to you during the relevant activities. **Durations of activities are estimations and may vary.

Your participation in this study is voluntary, and you may withdraw at any time without repercussions. All information will be kept confidential and used only for academic research. Personal details will not be identifiable in any reports or publications. You have the right to review and discuss report content before finalization. For questions or more information, please contact the researcher. Your involvement is highly appreciated and contributes to our research.

By signing, you confirm your understanding and agreement of the terms stated above.

Organization or Project Name: _____

Name and role: _____ _____

Signature: _____

Date: _____

Appendix B. Interview Procedure

Appendix B.1 Procedure for Product Owner Role

Introduction of interviewer and project

1. Background of the interviewer.
2. Experiment details.

Personal Experience in Role

1. How would you introduce yourself?
    a. Origin? Age? Educational background?
2. How long have you been in a role as Product Owner?
    a. Did you consider other roles in IT? How did you get into this position?
3. Could you share your experiences as a Product Owner in your current project?
4. How many years of experience do you have in IT, and specifically in Agile teams?

Current Project and Target Audience

5. Can you tell me about the project you are currently working on?
    a. What is its nature, scope, and primary objectives?
6. Who is the primary target audience for your current project (customers, employees, etc.)?
    a. How many individuals are part of the target audience? Can you provide a range?
    b. Are individuals from the primary audience involved in the development process?
    c. If yes, in what way do they influence the project's approach and goals?
7. What do your sprints look like in terms of effort? Hours of labor, sprint length, velocity, etc.
8. What project management tool do you use for the project? E.g. Jira, Azure DevOps, etc.

Experience with the INVEST Framework

9. Were you familiar with the INVEST framework before participating in this study?
    a. **If yes**, how did you come to learn about it?
    b. **If no**, skip to Question 13. The framework will be further discussed during the workshop.
10. Could you explain what each letter in the INVEST acronym stands for?
11. Have you applied the INVEST criteria in your projects?
    a. Please share any experiences or outcomes.
12. What benefits and challenges have you encountered using the INVEST framework?
    a. How could these challenges be effectively addressed?
13. What benefits and challenges do you expect during this experiment?

Other Frameworks for Enhancing User Stories

14. Are there other frameworks or approaches you have used for enhancing user stories?
    a. **If yes**, was it effectively implemented?
        i. **If yes**, how did you ensure it was?
        ii. **If not**, what did you struggle with? What would you do differently?
    b. **If no**, skip to Question 15.
15. Are there aspects of INVEST you (expect to) find particularly more or less effective?
16. Are there aspects of INVEST you (expect to) find particularly more or less difficult to implement?
17. Based on your experience, do you have suggestions to better implement a method like INVEST?
18. How does the implementation of INVEST, or frameworks other than INVEST, affect team dynamics and collaboration in your experience?
    a. Does it contribute to better team communication?
    b. Have you experienced resistance within your team when it comes to trying new methods?
        i. Do you feel the team is open to exploring new methods to improve quality?

Appendix B.2 Procedure for Developer Role

Introduction of interviewer and project

1. Background of the interviewer.
2. Experiment details.

Personal Experience in Role

1. How would you introduce yourself?
    a. Origin? Age? Educational background?
2. How long have you been in a role as a Developer?
    a. Did you consider other roles in IT? How did you get into this position?
    b. What platform or programming language are you specialized in?
3. Could you share your experiences as a Developer in your current project?
4. How many years of experience do you have in IT, and specifically in Agile teams?

Current Project and Target Audience

5. Can you tell me about the project you are currently working on?
    a. What is its nature, scope, and primary objectives?
6. Who is the primary target audience for your current project (customers, employees, etc.)?
    a. Are individuals from the primary audience involved in the development process?
    b. If yes, in what way do they influence the project's approach and goals?

Experience with the INVEST Framework

7. Were you familiar with the INVEST framework before participating in this study?
    a. **If yes**, how did you come to learn about it?
    b. **If no**, skip to Question 11.
8. Could you explain what each letter in the INVEST acronym stands for?
9. Have you seen INVEST being applied in projects you worked on?
    a. Please share any experiences or outcomes.
10. What benefits and challenges have you encountered in projects applying INVEST?
    a. How could these challenges be effectively addressed?
11. What benefits and challenges do you expect during this experiment?

Other Frameworks for Enhancing User Stories

12. Are there other frameworks or approaches you have encountered for enhancing user stories?
    a. **If yes**, was it effectively implemented?
        i. **If yes**, how did you ensure it was?
        ii. **If not**, what did you struggle with? What would you do differently?
    b. **If no**, skip to Question 13.
13. Are there aspects of INVEST you (expect to) find particularly more or less effective?
14. Are there aspects of INVEST you (expect to) find particularly more or less difficult to implement?
15. Based on your experience, do you have suggestions to better implement a method like INVEST?
16. How does the implementation of INVEST, or frameworks other than INVEST, affect team dynamics and collaboration in your experience?
    a. Does it contribute to better team communication?
    b. Have you experienced resistance within your team when it comes to trying new methods?
        i. Do you feel the team is open to exploring new methods to improve quality?

Appendix B.3 Procedure for Reflection

Recap of Project

1. Recap of the study's goals and the significance of the participant's input.

Personal Reflection on the INVEST Framework Experience

2. How has your understanding of the INVEST framework evolved throughout the project?
3. Can you share a memorable experience where the INVEST criteria directly impacted your work?

Experience with User Stories Post-Experiment

4. In what ways have you noticed a change in the quality or clarity of user stories after applying the INVEST framework?
5. Could you provide specific examples of how INVEST-aligned user stories affected your approach to certain tasks?

Challenges and Limitations

6. What challenges did you encounter while going over the INVEST criteria during this project?
7. Were there any limitations of the INVEST framework that became apparent as the project progressed?

Comparison with Previous Methods

8. How does the INVEST framework compare to other methods you have used for enhancing user stories?
9. Do you have a preference between INVEST and the other methods you've experienced? Please explain your reasons.

Perceived Benefits and Value

10. What benefits, if any, have you perceived in your work due to the application of the INVEST framework?
11. Would you recommend the use of the INVEST framework for future projects? Why or why not?

Feedback Loop on the Experiment Process

12. Looking back at the experiment process, including workshops and refinement meetings, what feedback can you offer for improvement?
13. Were there any stages of the experiment that you found particularly beneficial or challenging?

Closing the Interview

14. Are there any additional comments or insights you would like to share that we have not covered?
15. How do you feel about the potential impact of the findings from this study on the project team?

Appendix C. INVEST Checklist

The INVEST Checklist:

Independent

- Can the story stand alone without requiring other user stories to be completed first?
- Are there any dependencies on external factors or other user stories?
- Can this story be developed, tested, and implemented independently?

Focus on managing dependencies effectively with regular communication. Consider splitting the story into smaller sub-requirements. If complete independence is impractical due to the complex project nature, ensure effective management of the identified dependencies.

Negotiable

- Is there flexibility for changes in scope or details within the story?
- Are there aspects of the story open to discussion and adaptation?

Encourage flexibility in the development process, allowing for changes in scope, approach, or priority before finalizing requirements. Avoid making user stories too detailed to maintain their negotiable nature.

Valuable

- Does the story provide clear value to the end-user or customer?
- Is the purpose of the story aligned with the project goals?
- Can the value be clearly articulated and understood by the team?

Reassess the user story to ensure it aligns with end-user needs and the overall objectives of the project. Clarify and articulate the value proposition to make the story valuable to the client.

Estimable

- Is there enough information to estimate the time and resources needed?
- Do team members understand the story well enough to provide an estimate?
- Are there any ambiguous parts that make estimation difficult?

Clarify the user story to ensure developers can reasonably estimate the effort, time, and resources required. If ambiguity exists, seek additional information or break down the story further for clearer understanding.

Small

- Can the story be completed within one Sprint?
- Is the scope of the story narrow and well-defined?
- If the user story is considered large, can it be split into smaller, independent user stories?

Apply user story-splitting techniques, such as horizontal (dividing user stories by architectural layers) or vertical splitting (focusing on a single function across layers). Aim for a balance in size to ensure manageability within a Sprint without making user stories trivial or inefficient.

Testable

- Does the story include specific acceptance criteria or acceptance tests?
- Are these criteria clear, measurable, and achievable?

Add specific acceptance criteria or tests to the user story. Ensure these criteria are clear, measurable, and achievable to verify the story's success through testing.

Appendix D. Interview Recordings

In accordance with our participants' confidentiality requests, access to our recordings is restricted. For further requests, please contact the authors.