

Exploring the Role of Generative AI for Improving and Optimizing Sprint Planning in Agile Development

UNIVERSITY OF TURKU
Department of Computing
Master of Science (Tech) Thesis
Software Engineering
July 2025
Md Sharifur Rahman

Supervisors:
Antero Järvi (University of Turku)
Oshani Weerakoon (University of Turku)

UNIVERSITY OF TURKU
Department of Computing

MD SHARIFUR RAHMAN: Exploring the Role of Generative AI for Improving and
Optimizing Sprint Planning in Agile Development

Master of Science (Tech) Thesis, 76 p., 7 app. p.
Software Engineering
July 2025

Generative AI is transforming how software teams approach project management. This thesis investigates the application of large language models, such as ChatGPT, in Agile sprint planning. It begins with a review of current research and identifies persistent challenges in traditional sprint planning, including subjective backlog selection, inconsistent story point estimation, and uneven task assignment.

To address these issues, a web-based tool called GenSP was developed. GenSP leverages GenAI APIs to generate sprint backlogs, estimate story points, break down user stories, and assign tasks. The tool was evaluated using real project data, and its effectiveness was further assessed through a survey of experienced Agile practitioners.

The results indicate that GenAI can enhance sprint planning by automating backlog refinement, story point estimation, and task breakdown. However, the study also highlights concerns regarding the need for human oversight, the handling of complex business logic, and the protection of sensitive data.

In summary, this research demonstrates that GenAI can make sprint planning more effective and efficient. The findings provide practical guidance for teams considering AI integration in Agile workflows and suggest directions for future research and development.

Keywords: Artificial Intelligence, Agile Methodology, Scrum, Generative AI, Sprint Planning, Project Management

Acknowledgement

I want to express my sincere gratitude to my supervisors, Antero Järvi and Oshani Weerakoon, for their valuable guidance, continuous support, and encouragement throughout this thesis journey. Their insightful feedback helped shape my research and significantly improved the quality of my work.

I am especially grateful to the University of Turku (Software Engineering) for providing access to the ChatGPT API, which was instrumental in developing the tool used in this research.

I am also thankful to all the participants who took the time to respond to my survey. Their experiences and thoughtful insights greatly contributed to this thesis's results.

Finally, I warmly thank my family and friends for their constant encouragement, patience, and unwavering support, which made this accomplishment possible.

Contents

Acknowledgements	1
1 Introduction	1
1.1 Problem Statement	2
1.2 Goals and Objectives	3
1.3 Research Questions	3
1.4 Methodologies	4
1.5 Use of Generative AI	6
1.6 Thesis Structure	6
2 Background	8
2.1 Fundamentals of AI	8
2.2 Overview Generative AI	9
2.3 Generative AI in Software Engineering	10
2.3.1 Code Development and Assistance	11
2.3.2 Design and Creativity	11
2.3.3 Software Testing and Quality	12
2.3.4 Requirements Engineering	12
2.3.5 Documentation	13
2.4 Agile Methodologies and Scrum	13
2.4.1 Agile Methodologies Overview	14

2.4.2	Scrum	16
2.4.3	The Significance of Scrum in Software Development	18
2.5	Limitations and Risks of Generative AI	18
3	Sprint Planning and Generative AI	20
3.1	Sprint Planning in Scrum	20
3.2	Challenges in Traditional Sprint Planning	21
3.3	Current Research on Generative AI in Sprint Planning	22
3.4	Research Gap and Thesis Motivation	26
4	Methodology	27
4.1	Research Design	27
4.1.1	Background Studies	27
4.1.2	Prototype Design and Development	28
4.1.3	Data Collection and Evaluation Strategy	29
4.1.4	Research Questions and Mapping to Methods	30
5	System Design and Implementation	33
5.1	Core Functionalities	34
5.1.1	Projects Management	34
5.1.2	Backlog Management	35
5.1.3	Story Points Estimation	36
5.1.4	Sprint Generation	37
5.1.5	Task Breakdown	37
5.1.6	Task Assignment	37
5.1.7	User Authentication and Management	38
5.2	GenSP System Architecture	38
5.2.1	Backend Implementation	39
5.2.2	Technologies and Tools	39

5.2.3	ChatGPT's API Integration	41
5.2.4	Frontend Implementation	43
5.3	Prompt Engineering	44
5.4	Challenges and Lessons Learned	47
6	Results and Discussion	49
6.1	Evaluation using MyFlavoria Project	49
6.1.1	User Story Breakdown and Estimation	50
6.1.2	Sprint Backlog Generation and Task Distribution	54
6.1.3	Overall analysis	56
6.2	Survey Results	59
6.2.1	Survey Demographics	59
6.2.2	Quantitative Results	61
6.2.3	Qualitative Results	67
6.3	Mapping Research Questions with Results	70
6.4	Discussion	72
6.4.1	Evaluating GenSP Tool Data	72
6.4.2	Interpretation of Survey Results	73
6.4.3	Key Contribution	73
6.4.4	Limitations	74
7	Conclusion	75
7.1	Future Work	76
	References	77
	Appendices	
A	Survey Questions	A-1
A.1	Generative AI in Sprint Planning: Survey Questions	A-1

List of Figures

- 2.1 Relation among AI, ML, DL and GenAI [18]. 10

- 5.1 GenSP Product Backlog. 35
- 5.2 GenSP Story Point Generation. 36
- 5.3 GenSP System Architecture. 39

- 6.1 GenAI Benefits in Sprint Planning (Survey Results) 67

List of Tables

6.1	User Stories, Subtasks, and Estimated Points	50
6.2	GenSP Suggested Sprint and Task Distribution	54
6.3	Participant's current company location	59
6.4	Participant's software development experience	60
6.5	Familiarity with Scrum	60
6.6	Survey Results on GenAI in Sprint Backlog Generation	62
6.7	Survey Results on GenAI in Story Points Estimation	63
6.8	Survey Results on GenAI in Story/Task Breakdown	65
6.9	Survey Results on GenAI in Task Distribution	66

1 Introduction

Artificial Intelligence (AI) is rapidly reshaping modern software engineering, driven significantly by recent advancements in Generative AI (GenAI). Large language models (LLMs) like ChatGPT, developed by OpenAI, have recently attracted significant attention due to their impressive performance on various tasks, gaining immense popularity among early adopters—some of whom describe them as disruptive technologies across many domains [1]. Contemporary GenAI tools now support software developers in code generation, suggesting optimal code refactoring [2], automatically generating and validating test cases [3], and assisting in documentation [4]. Given these capabilities, new opportunities exist in exploring the integration of GenAI within Agile methodologies, aiming to improve the efficiency, accuracy, and overall effectiveness of Agile practices.

Agile software development has gained widespread industry adoption in response to the demand for faster, cost-effective solutions in dynamic environments, with most companies opting for agile due to its highly result-oriented approach [5]. So, different Agile methodologies like Scrum and Kanban have become a standard approach in software industries. Scrum is the most widely adopted Agile methodology, used not only in software development but also in various other domains [6]. It structures development into short, focused cycles called sprints, which produce working software features at regular intervals. This enables teams to incorporate feedback and adjust to evolving requirements rapidly.

Sprint planning, as described in the Scrum Guide, is a collaborative activity that initiates each sprint by bringing the entire Scrum Team together to lay out the work to be performed [7]. During this session, the team reviews and selects the most relevant user stories from the product backlog based on the sprint goals. The selected items are then estimated in terms of complexity and effort, often using a point-based system. To ensure clarity and effective execution, these stories are further broken down into smaller, actionable tasks. Finally, tasks are distributed among team members according to their expertise, availability, and capacity. The sprint planning ensures everyone is aligned on what can be delivered and how, with input from the product owner, developers, and scrum master, and is timeboxed based on sprint length to maintain focus and efficiency [8].

1.1 Problem Statement

In Scrum, Sprint Planning is a crucial team-oriented activity that defines the objectives, scope, and deliverables for each development iteration [9]. Effective planning is vital for the success of any software project. However, despite its significance, traditional sprint planning methods face several inherent challenges [10] [11] [12] [13].

One major issue is the heavy reliance on subjective and heuristic estimates from team members, which frequently leads to inconsistencies and inefficiencies [10] [11]. Bias and subjectivity in these estimates can result in unrealistic or unclear sprint goals and inaccurate effort estimation. Such problems often cause overcommitment or underutilized team capacity, both of which are primary contributors to sprint failures and diminished business value [12].

Moreover, traditional approaches often do not provide systematic methods for identifying risks and dependencies early in the planning process [11]. When user stories are manually broken down into tasks, important subtasks or dependencies

can be overlooked, leading to bottlenecks. Additionally, errors in task assignment may produce uneven workloads, where some team members are overloaded while others are underutilized. Sequential task dependencies further contribute to idle time, which becomes especially inefficient in larger projects [13].

1.2 Goals and Objectives

This thesis investigates how GenAI can enhance sprint planning by addressing key limitations found in traditional methods. The research specifically evaluates the impact of GenAI across multiple, interconnected aspects of the sprint planning process. The main objectives are as follows:

1. The effectiveness of GenAI in sprint backlog creation with user stories align with sprint goals
2. The usefulness of GenAI in estimating user story points and supporting team consensus during sprint planning
3. The effectiveness of GenAI in breaking down user stories into actionable tasks
4. The helpfulness of GenAI in suggesting skill-based task assignments and balancing team workloads during sprint planning
5. The overall GenAI's impact on the effectiveness, efficiency, and quality of sprint planning.

1.3 Research Questions

To address these objectives comprehensively, this thesis is guided by the following research questions:

Research Question 1 (RQ1) : What is the value and effectiveness of GenAI in

sprint backlog creation, specifically in suggesting relevant user stories and identifying missing details or dependencies?

Research Question 2 (RQ2): How effective is GenAI in estimating user story points during sprint planning in terms of accuracy, facilitating team consensus, and serving as a useful starting point for team discussions compared to traditional estimation methods?

Research Question 3 (RQ3): How effective is GenAI in breaking down user stories into actionable tasks, identifying task dependencies, and providing a useful starting point for task refinement in sprint planning?

Research Question 4 (RQ4): How helpful is GenAI in suggesting task assignments based on individual skills, experience, or development goals, and in balancing workloads among team members during sprint planning?

Research Question 5 (RQ5): How does GenAI contribute to improving the overall effectiveness, efficiency, and quality of the sprint planning process?

1.4 Methodologies

This thesis adopts a mixed-methods approach, combining design science research (DSR) [14] with empirical survey [15] analysis to investigate the potential of GenAI in Agile sprint planning. The research is driven by two main focus areas:

1. Developing a web-based tool that leverages GenAI for key sprint planning tasks
2. Evaluating its practical value based on feedback from Agile practitioners.

The research process began with a comprehensive literature review. This review helped identify persistent challenges in traditional Agile sprint planning, such as difficulties in aligning sprint backlog items with defined goals, subjective and inconsistent effort estimation, and issues with workload distribution. Insights from

the literature also highlighted a gap in practical research on integrating GenAI into sprint planning, motivating the development of a novel AI-supported solution.

To address these challenges, a functional prototype named GenSP is designed and developed. This tool supports several core sprint planning activities: generating sprint backlogs that align with sprint goals and team capacity, estimating story points from user story descriptions, breaking down user stories into actionable tasks, and distributing tasks among team members. GenSP is built as a full-stack web application, using React.js for the frontend, Express.js for the backend, and MongoDB for data storage. The tool integrates OpenAI's GPT-4 model as the core GenAI engine, with prompt engineering techniques such as Chain-of-Thought (CoT) [16] used to encourage logical, well-structured responses.

To evaluate the tool, its features were tested using the backlog of a real-world software project. The focus was on assessing the completeness, relevance, and alignment of the AI-generated outputs with Agile practices. At this time it was not feasible to involve a live Scrum team; this demonstration was complemented by a structured survey distributed to experienced software developers working with Agile methodologies. The survey included Likert-scale questions to assess perceived usefulness and accuracy, scenario-based items for direct feedback on AI outputs, and open-ended questions for qualitative insights.

Before participating in the survey, respondents were introduced to the GenSP tool through a video walkthrough or live demo. This ensured all participants had a consistent understanding of the tool's purpose and features, making their responses more meaningful for analysis.

Overall, this methodology provides a rigorous foundation for evaluating how GenAI can improve sprint planning, balancing theoretical research with practical implementation and real-world feedback.

1.5 Use of Generative AI

In this thesis, GenAI was used for limited and specific editorial purposes. OpenAI's ChatGPT provided strictly grammatical and structural assistance, improving sentence clarity, correcting language errors, enhancing document flow, and refining academic tone. GenAI-processed content underwent careful verification to ensure originality, coherence, and strict adherence to scholarly standards.

AI was not used in core intellectual contributions, including research design, literature analysis, prototype development, evaluation methodology, and results interpretation.

1.6 Thesis Structure

This thesis is organized into six chapters, each designed to address specific aspects of the research objectives and provide a logical flow of discussion:

Chapter 2: Background - This chapter reviews the basics of AI and discusses how GenAI has developed and where it is applied. The chapter also covers software project management, with a focus on Agile methods like Scrum, and points out the risks and limitations of using GenAI.

Chapter 3: Sprint Planning and Generative AI - This chapter explains how sprint planning works in Scrum. It outlines problems in traditional planning and examines how GenAI can help. The chapter also reviews recent research, highlights the research gap, and explains the motivation for this thesis.

Chapter 4: Methodology - The methodology chapter describes how the research was designed and carried out. It explains the use of design science research, the development of the GenSP prototype, and how data was collected and evaluated. The chapter also shows how each research question relates to the chosen methods.

Chapter 5: System Design and Implementation - This chapter details how

the GenSP tool was built. It covers its main features, including backlog management, story point estimation, and task assignment. The technical stack, backend and frontend design, and integration with GenAI are described. Challenges during development and key lessons learned are also discussed.

Chapter 6: Results and Discussion - This chapter presents the results of testing GenSP with real project data and survey feedback from Agile professionals. It includes both quantitative and qualitative findings. The impact of GenAI on sprint planning is analyzed, the research questions are answered, and study limitations are discussed.

Chapter 7: Conclusion - The final chapter summarizes the main findings and contributions of the thesis. It reflects on what the research means for practice and suggests areas for future work.

This structure guides readers step by step, from the initial problem to the final conclusions, and shows how GenAI can support better sprint planning in Agile teams.

2 Background

In recent years, GenAI has become an important and influential force in software engineering. AI-powered tools like GitHub Copilot are reshaping software development by offering real-time code suggestions that boost productivity and reduce cognitive load. [17]. These tools showcase how GenAI can be applied to real development work and highlight its potential to support many stages of the software engineering process, particularly in Agile methodology.

This chapter provides the foundation for understanding GenAI's impact on software engineering and Agile practices. It introduces core concepts in AI, Machine Learning (ML), and Deep Learning (DL), then explores the growing role of GenAI in software tasks such as coding, design, testing, and documentation. The chapter also covers Agile methodologies, with a focus on Scrum and the challenges of sprint planning.

2.1 Fundamentals of AI

AI is a branch of computer science that seeks to understand and replicate intelligence by drawing on concepts from logic, mathematics, probability, learning, perception, and action [18]. The field aims to develop autonomous systems capable of functioning in dynamic and uncertain environments [19].

The origins of AI trace back to the 1950s, when Alan Turing introduced the idea of machine intelligence and proposed the "Turing Test" as a way to evaluate

it [18]. Since then, advances in technology and data have contributed to the rapid development of AI, which is now applied in areas such as language processing, image recognition, autonomous vehicles, and healthcare.

ML, a key area within AI, enables systems to learn from data and adapt without explicit programming [20]. Reviews such as that by Mian et al. [21] describe how ML is applied in healthcare for disease diagnosis and personalized treatments, in e-commerce for chatbots and recommendations, in finance for fraud detection, and in sentiment analysis of social media and text data.

DL, a specialized subset of ML that uses Artificial Neural Networks (ANNs), is fundamental to many modern AI systems, and it is commonly used in image and speech recognition, Natural Language Processing (NLP), and autonomous driving [20]. Surveys have documented DL's diverse applications, including automating medical diagnostics, improving energy-load forecasting, enhancing text classification in NLP, signal de-noising, and computer vision tasks. Challenges related to limited data and computational requirements are also noted in the literature [22].

2.2 Overview Generative AI

GenAI refers to systems that use generative models to create new text, images, or other media by learning patterns from existing data [23]. In contrast to traditional AI, which primarily focuses on pattern recognition and predictive analysis, GenAI models are capable of creating original outputs that often mirror work produced by humans. Major tech companies like Google, Meta, Microsoft, and OpenAI have developed advanced generative AI technologies, with OpenAI notably releasing DALL-E, a model capable of creating images from text descriptions using a variant of the GPT-3 architecture [24]. GenAI systems are not limited to creative tasks; they are increasingly being used as intelligent assistants that support knowledge work and answer complex questions [25], thereby reshaping how both individuals

and organizations approach problem-solving and productivity.

GenAI relies on advanced DL architectures, primarily Deep Generative Models (DGMs) such as Transformers, Latent Diffusion Models, VAEs, and GANs, each offering distinct approaches to modeling and generating data [26]. Recent advances, including multimodal integration and scalable Transformers, have enabled applications like text-to-image synthesis, though challenges like fuzzy VAEs and mode collapse in GANs remain [27].

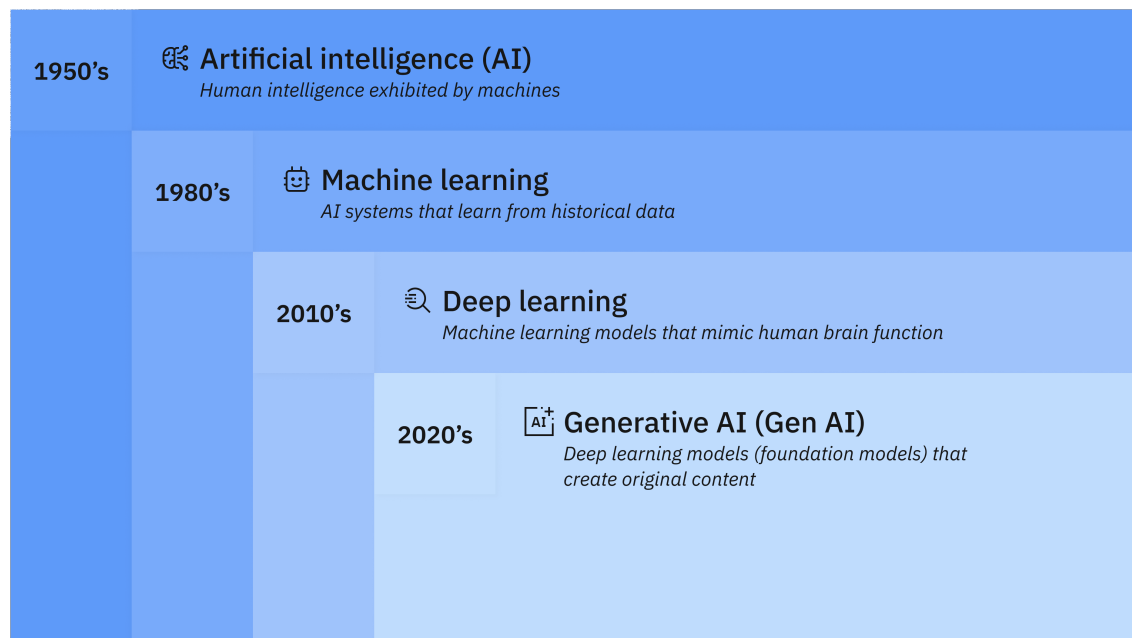


Figure 2.1: Relation among AI, ML, DL and GenAI [18].

2.3 Generative AI in Software Engineering

GenAI is rapidly gaining traction in software engineering. According to the Developers Survey of Stack Overflow 2024, 76% of respondents use or plan to employ AI tools in their development processes [28]. This suggests that the use of GenAI technologies by software professionals is expanding. Ebert et al. [4] provides a practical, practitioner-focused overview of how GenAI impacts software engineering, and it is

discussed here.

2.3.1 Code Development and Assistance

GenAI tools are increasingly vital in software engineering, automating code generation for a range of development tasks [4]. Popular AI-based code assistants such as GitHub Copilot, Tabnine, ChatGPT, and Google Bard leverage ML and NLP to provide real-time code suggestions and method implementations, offering valuable support to developers, though they rarely produce fully correct, ready-to-use code [2].

Evidence suggests that GenAI significantly enhances developer productivity. For instance, developers using GitHub Copilot completed an HTTP server task 55.8% faster than those without GenAI assistance [29]. Advanced models like GPT-4 can generate entire functions or classes from natural language prompts and solve coding problems from docstrings at or above human-level performance [4].

Modernizing legacy rule-checking systems is essential because they rely on manual expertise, lack adaptability, cannot express complex logic, and are limited to handling only simple, explicit requirements [30]. In addition, manual modernization is slow, error-prone, and loses logic intent [31]. A study shows that GenAI modernizes legacy systems by automating manual tasks, handling ambiguous logic, and enabling scalable, executable code generation [30].

2.3.2 Design and Creativity

GenAI is now making a noticeable impact in the creative and design areas of software engineering. GenAI-powered design tools can automate tasks, boost creative idea generation, and produce more unconventional outputs [32].

GenAI also supports more technical design tasks, such as chip layout and architectural planning. For example, in chip design, AI tools suggest circuit arrangements

that can boost performance [4]. In architecture, they help propose different building layouts for review. Overall, GenAI not only saves time but also encourages innovation by offering fresh design options and efficient solutions [4].

2.3.3 Software Testing and Quality

Software testing and quality assurance are vital steps in software development. Recent advances in GenAI, especially LLMs, enable dynamic test case generation, vulnerability prediction, adaptation to software changes, and reduced reliance on human testers [33]. This indicates that AI tools can read requirement documents or user stories and quickly turn them into test cases. This approach helps cover more scenarios and reduces the risk of missing important tests.

GenAI can automate the generation of unit and system test cases, craft test assertions, visualize call graphs, execute and report tests, analyze bugs, support debugging and automated repair, and document testing rationale [3]. By running these tests automatically, AI tools help teams find problems faster and keep the software reliable. GenAI can produce synthetic data for rare or edge cases. It helps test and validate autonomous systems in fields like medicine and aerospace [4]. Overall, GenAI makes testing more thorough and efficient, saving both time and effort.

2.3.4 Requirements Engineering

GenAI has begun to play a valuable role in improving the requirements engineering process, which is a critical early stage in software development. LLMs can automate and improve requirements engineering tasks by processing natural language, generating documentation, and simulating stakeholder perspectives [34]. This also indicates that GenAI-powered NLP tools can analyze requirements written in natural language to identify ambiguities, missing details, or even conflicting requirements

within project documentation. By flagging vague or inconsistent statements, these systems help teams clarify expectations early on, which minimizes misunderstandings and reduces costly changes later in the project.

Notably, recent research [35] shows work on automating requirements engineering in agile development using LLMs explores the use of GenAI on a larger scale. The paper addresses the labor-intensive process of converting requirements documents into user stories and test cases, proposing the tool GeneUS with the novel Refine and Thought (RaT) prompting technique. The study positions itself as the first implementation-level effort to automate user story generation with integrated test specifications [35].

2.3.5 Documentation

A major use of GenAI is generating, modifying, and improving text. This is especially useful for managing project documentation. GenAI tools can quickly summarize long documents and meeting notes, helping teams grasp important details and focus on higher-priority tasks [4]. Many older codebases lack clear documentation, but GenAI can help fill these gaps by creating documentation for legacy projects. Automated documentation tools powered by AI also make it easier to keep project records current. This not only improves code readability for the team but also helps new developers understand the system more quickly. Overall, the integration of GenAI in documentation ensures that knowledge is accessible, accurate, and easy to maintain throughout the development lifecycle.

2.4 Agile Methodologies and Scrum

Software project management guides the entire development process, ensuring efficient use of time, money, and resources. The following sections explore agile method-

ologies in more detail, with a particular focus on Scrum.

2.4.1 Agile Methodologies Overview

Some traditional software development models rely on strict planning and lengthy development cycles. These rigid approaches often struggle when customer requirements change frequently. In contrast, Agile methodologies offer flexibility to adapt quickly to changing requirements. Agile methods effectively handle uncertainty by emphasizing the skills of team members and the importance of their collaboration in software development [36]

The foundation of Agile lies in the Agile Manifesto, and it was introduced in 2001 by a group of software professionals seeking to improve development processes [37].

- Customer satisfaction through early, continuous delivery
- Embrace changing requirements
- Deliver working software frequently
- Business and developers collaborate daily
- Build projects around motivated people
- Communicate face-to-face
- Working software is the main progress measure
- Sustainable development pace
- Continuous attention to technical excellence
- Simplicity—maximize work not done
- Self-organizing teams produce best results

- Regular reflection and adjustment

Agile methodology organizes work into short, repeatable cycles known as iterations or sprints. Within each iteration, the team plans, develops, tests, and delivers a working increment of the product. Customer and stakeholder feedback obtained regularly is then used to refine goals and priorities for future iterations. Effective communication is critical to success, and agile methods cannot thrive without a foundation of strong teamwork and collaboration [38].

Agile methodology provides a flexible, people-focused approach to managing software projects. It emphasizes close teamwork, continuous improvement, and regular stakeholder involvement. Several Agile frameworks, such as Scrum, Kanban, and Scaled Agile Framework (SAFe), help teams apply these principles in practice. Scrum is the most widely used agile methodology, valued for its simplicity and adaptability across diverse contexts [6]. It uses short, fixed-length cycles called sprints and involves clearly defined roles and regular ceremonies like planning sessions and retrospectives. Kanban is an agile method initially developed by Toyota in the 1950s, emphasizing efficiency through visual workflow management, limiting work-in-progress (WIP), and managing flow [39]. SAFe scales agile practices for large enterprises and is structured around four levels—Team, Program, Portfolio, and Value Stream—featuring Agile Release Trains, Program Increments, specialized roles, and core values of quality, transparency, alignment, and program execution [40].

Every Agile framework focuses on collaboration, adaptability to changing requirements, and delivering incremental value. The choice among these frameworks depends on organizational size, project characteristics, and team-specific needs.

2.4.2 Scrum

Scrum is an agile framework that enables teams to work collaboratively, delivering value incrementally through short cycles and continuous feedback [7]. It provides a flexible structure that supports ongoing learning and improvement while allowing teams to adapt practices to best fit their specific needs. [7].

Roles in Scrum are designed to ensure a balanced and self-organizing team structure. A Scrum Team is a small, cross-functional, and self-managing group consisting of:

- Product Owner
- Scrum Master
- Developers

The Scrum Master, Product Owner, and Developers are focused on achieving the product goal [9]. The product owner serves as the voice of the customer and stakeholders, setting priorities in the product backlog so the team can focus on the most valuable work. The Scrum Master guides the team, supports Scrum practices, and helps solve problems that might slow down progress. The development team includes professionals who plan, create, and test each part of the product during a sprint. These teams bring together all the skills needed and are trusted to organize and manage their own work.

Artifacts in Scrum provide transparency and help track progress. The artifacts in Scrum are

- Product Backlog
- Sprint Backlog
- Increment

The product backlog is a list of all desired features and changes for the product, managed by the product owner. The Sprint Backlog contains the specific tasks and user stories the team commits to complete in the current sprint. The increment is the sum of all completed work at the end of a sprint, representing potentially shippable functionality [9].

Scrum ceremonies are regular meetings that help organize the team's work and keep everyone on the same page. The Scrum ceremonies are [9]:

- The Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Sprint planning begins each sprint by letting the team choose what tasks to tackle and how to approach them [7]. The daily scrum is a brief meeting where team members update each other on progress, talk about any challenges, and plan their next steps. At the end of the sprint, the Sprint Review allows the team to show their finished work to stakeholders and gather feedback. The sprint retrospective is a time for the team to look back on the sprint, discuss successes and areas for improvement, and decide how to work better in the future [9].

Scrum's roles, meetings, and tools help teams stay organized and flexible. This structure makes it easier for teams to deliver value often, adjust to changes quickly, and find ways to get better with each sprint.

2.4.3 The Significance of Scrum in Software Development

Scrum is now one of the most popular frameworks in the software industry, and it is the most widely used agile framework, accounting for over half of all agile practices reported in use [41]. Its clear structure, adaptability, and focus on teamwork make it a top choice for companies of all sizes, from startups to large corporations. Surveys show that many software teams use Scrum and value its positive effects on productivity, quality, and team morale.

A major reason for Scrum's success is its ability to handle changing requirements and customer feedback. By using short sprints and regular check-ins with stakeholders, teams can deliver value quickly and adjust to new project needs. This approach helps speed up delivery and lowers risks in complex projects. Scrum ceremonies (e.g., sprint reviews) provide continuous stakeholder feedback, creating a powerful way to develop products iteratively [6].

Scrum also encourages self-organizing teams and ongoing improvement. Developers begin each day with stand-up meetings to share updates and plan tasks. Rather than having a project manager, Scrum teams are guided by a Scrum Master and coordinate work through a Product Owner, who manages the backlog [6]. Many organizations find that using Scrum improves communication, boosts employee engagement, and supports better problem-solving. Its regular meetings and clear tools make it easier for everyone to see progress and stay aligned [41].

2.5 Limitations and Risks of Generative AI

GenAI has rapidly advanced in recent years, enabling machines to create text, code, images, music, and even videos with remarkable accuracy. However, despite their potential benefits, GenAI systems come with significant risks and limitations.

Beltran et al. [42] explored that GenAI presents several risks for public sector

organizations. One of the main concerns is the potential leakage of confidential information, as GenAI tools may unintentionally reveal sensitive government data. Another significant risk is bias and discrimination, since these systems can carry over unfair patterns from their training data, which may result in unjust decisions. GenAI can also introduce cybersecurity weaknesses, making organizations more vulnerable to data breaches [42].

Additionally, these tools are sometimes known to produce convincing but false information, a problem called hallucination [43]. Other risks include high energy use, which impacts the environment, and the possibility that heavy reliance on GenAI could lead to reduced critical thinking and accountability within public sector teams [42].

Over-dependency on GenAI can lead to diminishing creativity and ability to generate original thoughts or designs [44]. The research of Abbas et al. (2024) found that the use of GenAI tools like ChatGPT among university students brings several risks to education. Frequent use is linked to increased procrastination, as students may delay tasks when they rely on AI for support. Over time, depending too much on these tools can weaken students' memory and cognitive abilities [45].

3 Sprint Planning and Generative AI

This chapter highlights recent research on Agile sprint planning, identifies existing gaps in the current literature, and explains the motivation for exploring the potential of GenAI to address these challenges.

3.1 Sprint Planning in Scrum

Sprint planning is a key ceremony in Scrum that marks the start of each sprint. During this meeting, the Scrum team collaboratively decides what tasks will be completed in the upcoming sprint and how to approach them [46]. The main purpose is to create a clear, realistic plan and ensure everyone shares a common understanding of the sprint's goals.

All Scrum team members have distinct roles during sprint planning. The product owner defines the sprint goal and relevant backlog items, while the Scrum team selects tasks to complete and plans how to achieve them within the sprint [47]. The product owner prioritizes and clearly explains the items from the product backlog, and his/her objective is to add more value to the product in the current sprint. The Scrum Master guides the meeting, helps resolve issues, and ensures the team follows Scrum practices. The development team assesses tasks and breaks them into manageable pieces. Agile teams use various estimation methods like Planning Poker (Fibonacci-based cards), T-Shirt Sizing (abstract XS–XL sizes), Dot Voting (stakeholder prioritization), Bucket System (grouped story sizes), Large/Uncertain/Small

(minimal categories), Ordering Method (collaborative ranking), and Divide Until Max Size (splitting oversized stories) to assess effort and prioritize backlog items efficiently [48]. the most widely used project management tools in Agile software development include Jira, Trello, Asana, Microsoft VSTS, and Pivotal Tracker, due to their strong support for Scrum and Kanban practices, customization, and team collaboration features [49].

At the end of sprint planning, the result is a sprint backlog, which includes the sprint goal, chosen items, and the plan to deliver them. Sprint planning is limited in time, taking up to eight hours for a month-long sprint, and is shorter for shorter sprints [46]. These outcomes provide a solid foundation, ensuring the team stays organized, aligned, and focused throughout the sprint.

3.2 Challenges in Traditional Sprint Planning

Traditional sprint planning plays a central role in the Scrum framework, but it frequently brings about a number of challenges for software development teams. Biases and gaps in knowledge among team members can create various problems, often making the sprint planning process less effective.

A typical product backlog is often large, complex, and constantly evolving. Traditional sprint planning often relies on subjective, heuristic estimates, causing inconsistencies and inefficiencies [10] [11]. Setting a clear sprint goal is essential, as studies show that teams without a well-defined objective struggle to stay focused and achieve their targets [50]. Without careful planning, important backlog items that are critical to the sprint goal may be overlooked, which can undermine the success of the sprint

Estimation bias can lead to unclear goals, inaccurate effort predictions, and either overcommitting or underutilizing team capacity [10] [11]. Common methods like T-shirt sizing, Planning Poker, Dot-voting, and the Ordering Rule are popular

among agile teams for estimating user story complexity [48]. While they are simple and foster collaboration, they remain subjective and prone to bias [10]. These estimates are usually based on experience and gut feeling rather than data, which can lead to either overcommitting or underutilizing team capacity. Teams often struggle to accurately estimate the effort required for backlog items. Inexperience with estimation techniques or having estimates imposed by management can undermine team self-organization. This often leads to unrealistic commitments, missed deadlines, and increased technical debt [51].

Lack of systematic methods for identifying risks and dependencies, along with manual task assignment, can lead to overlooked tasks, project bottlenecks, uneven workloads, and increased idle time, especially in larger projects [11] [13].

Additionally, lack of shared understanding of project goals and priorities often leads to misinterpretations, rework, and delays, especially in distributed teams where ambiguous requirements cause teams to operate on differing assumptions [52].

Overall, these challenges make it difficult for teams to plan effectively and deliver consistent value. Addressing these issues is essential for improving team performance and achieving better results in Agile projects.

3.3 Current Research on Generative AI in Sprint Planning

In recent years, research on GenAI has grown significantly within the fields of Agile software development and Scrum. However, the specific application of GenAI to sprint planning remains largely underexplored. There is considerable opportunity for further investigation in this area, as only a limited number of studies have addressed how GenAI can be used to improve and optimize the sprint planning process. This gap highlights the need for more focused research to better understand and realize

the potential benefits of GenAI in Agile sprint planning.

Prior research shows that it has been researched for some time to integrate AI into the agile methodology. Dam et al. [53] propose that artificial intelligence can fill this gap by automating tasks such as backlog refinement, sprint planning, effort estimation, and risk management. Their framework outlines how AI engines can analyze both structured and unstructured project data to support better decision-making throughout Agile processes. However, they also emphasize that human judgment and collaboration remain essential, and AI should be viewed as a tool to support, not replace, Agile teams [53].

The research of Bahi et al. [54] highlights that GenAI technologies—such as ChatGPT, GitHub Copilot, and Tabnine—are now helping Agile teams by automating tasks like code generation, testing, documentation, and backlog refinement. GenAI also supports better decision-making, improves team collaboration, and offers data-driven insights for sprint planning and progress tracking. While the integration of GenAI can accelerate delivery, foster innovation, and improve software quality, the paper also notes certain limitations. These include the need for human oversight, ethical concerns, and the importance of context awareness. The authors conclude that GenAI should be seen as a tool to support Agile teams rather than replace them, and they encourage ongoing research and practical evaluation to make the most of AI's benefits while managing its risks [54].

Ostrowski et al. [55] highlighted that the specific integration of GenAI in IT project management remains underexplored. GenAI offers significant potential for automating routine tasks, optimizing resource allocation, and supporting data-driven decision-making, but its adoption is challenged by issues such as data quality, algorithmic bias, and organizational resistance. Despite proven benefits in other industries, there is a notable lack of empirical research and structured frameworks for deploying GenAI within established project management methodologies like Agile

and Scrum [55].

Several research studies have explored user story point estimation using GenAI. Prior to the rise of GenAI, Choetkiertikul et al. [56] applied novel deep learning models to estimate user story points. The proposed models mark a significant advance by automatically learning complex semantic patterns from raw user story texts without the need for manual feature design. Their model, which combines Long Short-Term Memory (LSTM) and Recurrent Highway Networks, outperforms established baselines and alternative machine learning methods across a large and diverse dataset, demonstrating the strong potential of deep neural networks to improve the consistency and accuracy of story point estimation in Agile software development [56].

Mallidi et al. [57] investigate how GenAI tools such as ChatGPT and GitHub Copilot can improve Agile story point estimation and overall software development productivity. Their findings show that GenAI tools can accelerate coding, enhance code quality, and support multiple programming languages, resulting in 15–30% time savings in various case studies. However, challenges remain around code consistency, legal compliance, and potential over-dependence on AI. To address these factors, the authors propose an adjustment model for story points that incorporates GenAI’s impact on efficiency and quality. They conclude that while GenAI offers clear productivity benefits, teams should ensure proper training and careful integration to mitigate associated risks [57].

Islam et al. [58] present a multimodal AI framework that integrates text, images, and categorical data to enhance story point estimation in Agile software development. Using models like BERT, CNN, and XGBoost, their approach predicts story points with greater accuracy, especially for simpler tasks, achieving up to 77% accuracy and a 0.84 F1-score without severity data. However, the study is limited by a small, imbalanced dataset and noise in image inputs. The authors suggest

that expanding the dataset, leveraging synthetic data, and fine-tuning models could further improve performance and address current limitations [58].

There are several research studies on GenAI usage in Agile methodologies; notably, Barcaui et al. [53] compared project plans made by a generative AI model and by an experienced human project manager, using a real-world mobile app project as a case study. Their results show that AI can quickly create clear and organized project plans, especially for basic tasks like outlining scope, estimating costs, and identifying risks. However, the plans created by the human project manager were usually more detailed and realistic, especially when it came to understanding the market, setting specific goals, planning resources, and dealing with complex issues. The study found that AI and humans each have unique strengths: AI is fast and consistent, but humans are better at creativity, contextual thinking, and handling uncertainty. The authors conclude that the best results come when project managers use AI as a tool, combining AI's efficiency with human expertise to create stronger, more practical plans [53].

Rahman et al. [35] work on automating requirements engineering, present a tool named "GeneUS," that leverages GPT-4 to automate the generation of user stories and associated test case specifications from requirements documents. The tool applies a novel prompt engineering technique called Refine and Thought (RaT), which improves the quality and clarity of outputs by refining and structuring LLM prompts. Through validation with software practitioners, the tool's output was found to be effective and highly rated in readability, understandability, and technical quality [35]. Adopting a similar approach, a tool to aid sprint planning could use LLMs to automatically analyze backlog items and requirements, generate well-formed user stories, estimate story points, and propose task breakdowns, thereby reducing manual effort and enhancing the accuracy and efficiency of Agile planning processes.

The research described above highlights that only a limited number of studies have directly addressed how GenAI can improve or optimize the sprint planning process. Despite the growing interest in applying GenAI within Agile software development, the use of GenAI in sprint planning remains largely underexplored.

3.4 Research Gap and Thesis Motivation

Recent research has looked at how GenAI can help in Agile software development and Scrum. However, most of these studies focus on general tasks like backlog refinement or story point estimation. Few have explored how GenAI can be used for the full sprint planning process. There are still many open questions about how GenAI can help teams choose backlog items, estimate effort, or spot risks during sprint planning.

So far, most findings show that GenAI can speed up some Agile activities and improve decision-making. But there is little evidence or clear frameworks for how to use GenAI in real sprint planning meetings. This leaves a gap in our understanding of how GenAI could make sprint planning more accurate and efficient.

This thesis is motivated by the need to fill that gap. It will investigate how GenAI tools can support different steps of sprint planning and how teams can best combine AI suggestions with their own experience. The aim is to provide practical evidence and guidance for Agile teams who want to use GenAI to improve their sprint planning outcomes.

4 Methodology

This chapter outlines the research methodology adopted in this thesis, which follows a Design Science Research (DSR) approach [14] to develop and evaluate GenSP—a GenAI-powered tool aimed at enhancing Agile sprint planning. GenSP supports sprint backlog generation, story point estimation, task breakdown, and task distribution, leveraging GPT-4 and prompt engineering. Its effectiveness was assessed using historical sprint data and a structured survey of Agile practitioners.

4.1 Research Design

DSR is well-suited for research that involves creating and evaluating artifacts to solve identified problems in practice. According to Hevner et al. [14], DSR aims to enhance human and organizational capabilities by creating innovative artifacts that provide solutions to real-world problems through iterative building and evaluation. This approach aligns with the goals of this thesis. This research addresses inefficiencies in traditional sprint planning by developing a novel GenAI-supported tool.

4.1.1 Background Studies

The research began with an extensive literature review to identify persistent challenges in Agile sprint planning and the latest research on using GenAI in sprint planning and relevant research gaps. As discussed in Chapter 3, common issues

include difficulties in aligning sprint backlog items with defined goals [50], as well as challenges in accurately estimating effort due to subjectivity and bias [10]. These problems have been shown to negatively affect team performance, reduce planning accuracy, and undermine overall sprint effectiveness. The motivation for this study arose from the growing potential of GenAI to address these inefficiencies by providing intelligent, data-driven support during the sprint planning process.

The primary objective of this thesis is to explore how GenAI can enhance the sprint planning process by addressing key challenges faced in traditional Agile practices. To support this objective, a functional artifact was developed with core features designed to assist critical sprint planning activities. The main functionalities of the tool include,

- Generating sprint backlogs aligned with sprint goals and team capacity.
- Estimating story points from user story descriptions and acceptance criteria.
- Breaking down user stories into actionable tasks.
- Distributing tasks among team members based on their available capacity.

These objectives directly aligned with the research questions and aimed to explore how GenAI could assist in improving sprint planning accuracy and efficiency.

4.1.2 Prototype Design and Development

To meet the defined objectives, a functional prototype named GenSP is designed and developed. It is a web application built using a modular, full-stack architecture comprising React.js for the frontend, Express.js for the backend, and MongoDB for data persistence. Each module in the system (e.g., projects, backlogs, sprints, users) was designed with separation of concerns and scalability and integration with existing project management tools in mind.

ChatGPT, developed by OpenAI, is a generative AI chatbot powered by LLMs, capable of engaging in human-like conversations and performing a wide range of text-based tasks [59]. In this thesis, ChatGPT (GPT-4) is used as the core GenAI engine to support intelligent decision-making in sprint planning. To enhance the quality and consistency of the AI's responses, prompt engineering techniques—particularly CoT prompting—were applied. CoT encourages the model to generate structured and logical reasoning steps, improving the reliability of outputs in complex planning scenarios [16].

The GenSP tool was developed to evaluate the feasibility of using GenAI to support specific sprint planning tasks, as described in Section 4.1.1. It serves as a proof of concept to assess whether GenAI can provide meaningful assistance in generating sprint backlogs, estimating effort, breaking down user stories, and distributing tasks within Agile workflows.

4.1.3 Data Collection and Evaluation Strategy

In the current stage, it is not feasible to evaluate the GenSP tool with an actual Scrum team during a live sprint cycle. Instead, the tool is tested using the backlog of an existing software project—the MyFlavoria web application [60]. This application's product backlog and historical sprint data were utilized to examine whether the tool's core functionalities—namely sprint backlog generation, story point estimation, task breakdown, and task distribution—could operate as intended. The evaluation focused on the qualitative aspects of the AI-generated outputs, particularly their completeness, relevance to the sprint goal, and alignment with Agile principles.

To supplement this demonstration, a structured survey was designed and distributed to software developers with practical experience in Agile methodologies. According to Kasunic et al. [15], "A survey is a data-gathering and analysis ap-

proach in which respondents answer questions or respond to statements that were developed in advance." This is the most valuable part of this thesis. The primary aim was to gather their perceptions regarding the usefulness, accuracy, and reliability of GenAI-powered sprint planning support, particularly across the four use cases implemented in the GenSP tool.

The survey consisted of three main types of questions:

- Likert-scale questions to assess perceived usefulness, accuracy, and trust in AI-generated outputs
- Scenario-based evaluation items, where participants reviewed and rated sample outputs produced by the tool
- Open-ended questions, which invited respondents to share qualitative feedback, reflections, and concerns about GenAI in Agile sprint planning.

A Likert scale is made up of multiple Likert-type items, where respondents rate their agreement on an ordered scale (e.g., Strongly Agree to Strongly Disagree); scores are numerically coded, summed or averaged, and can be analyzed with both parametric and non-parametric methods [61].

Before completing the survey, participants were introduced to the tool and its capabilities through either a brief video walkthrough or a live demonstration session. This ensured that all participants had a clear understanding of the tool's purpose, features, and user flow, allowing for more informed and meaningful responses.

4.1.4 Research Questions and Mapping to Methods

Five main research questions mentioned in section 1.3 guide this thesis, each focusing on a key area where GenAI could support and enhance Agile sprint planning. The approach to answering each question combines artifact demonstration, empirical evaluation, and practitioner feedback to ensure technical and practical relevance.

To address RQ1, the GenSP tool was designed to automate sprint backlog creation by analyzing sprint goals, team capacity, and available backlog items. The effectiveness of these AI-generated suggestions was evaluated through both hands-on demonstration using the Flavoria project’s sprint data and survey questions asking practitioners to assess the value and relevance of the AI’s outputs.

For RQ2, GenSP generates story point estimates for user stories based on their descriptions and acceptance criteria. These estimates were reviewed in the demonstration phase and rated by survey participants for perceived accuracy and usefulness, with additional feedback gathered on how GenAI may influence estimation discussions within teams.

To investigate RQ3, the tool provides automatic task breakdowns for user stories and highlights potential dependencies. Both the demonstration and the survey explored the clarity, completeness, and practicality of these task breakdowns, as well as participant comfort in using GenAI outputs as a starting point for further refinement.

RQ4 is examined by analyzing the tool’s capability to recommend task assignments that account for team capacity and expertise. The demonstration compared GenSP’s automated task assignments to historical team practices, while the survey collected practitioner opinions on the usefulness and fairness of AI-driven task distribution.

This overarching question, RQ5 is answered through a combination of artifact evaluation and practitioner feedback. The demonstration phase assessed GenSP’s practical outcomes, while the survey gathered broader insights on how GenAI affects planning quality, team satisfaction, and sprint outcomes.

In summary, each research question is addressed through a combination of tool-based demonstration (aligned with the DSR methodology) and practitioner feedback (via surveys). This dual approach ensures both practical validation and empirical

evaluation of GenAI's role in Agile sprint planning.

5 System Design and Implementation

This chapter presents the development process and architectural details of GenSP which is an AI-powered application.

To build this application, several modern technologies were combined. The backend is built with Express.js, which handles all the server-side logic and data management. The frontend uses React.js to create a user-friendly interface where team members can interact with sprint planning tools and view AI-generated suggestions. MongoDB is used as the database to store all relevant project data, including product backlogs, sprints, tasks, and user profiles. Most importantly, the application is connected to the ChatGPT API, which allows the application to generate user stories, estimate story points, and break down tasks.

This chapter details the architecture, design choices, and technical implementation of the system. It explains how each component was developed and integrated to provide a seamless user experience and ensure the reliability and security of the application. By documenting the development process, this chapter lays the foundation for evaluating the practical benefits and limitations of GenAI in real-world sprint planning scenarios.

5.1 Core Functionalities

At the start of developing GenSP, the initial scope and requirements were fairly modest. The main objective was to create basic REST (Representational State Transfer) APIs to facilitate core sprint planning tasks like sprint backlog management and story point estimation. A RESTful API is a standardized interface that enables secure and efficient communication between computer systems over the internet [62]. However, as the project progressed, it became evident that these basic features were not sufficient. The application needed additional capabilities to fully support the complexities of GenAI-powered Agile project management. This realization led to the inclusion of more features, expanding the system to better address the needs of users and deliver a more comprehensive solution.

Ultimately, GenSP evolved into a comprehensive platform featuring several interconnected backend modules, moving well beyond the initial vision of a simple API. Each module was carefully designed to integrate with others, resulting in a seamless user experience that supports a wide range of Agile project management tasks. While this broader scope introduced additional complexity to both the backend and frontend, it was essential to deliver a tool that truly meets the needs of Agile teams. As a result, GenSP now offers not only advanced GenAI-driven capabilities for sprint planning and suggestions but also a robust set of core project management features. The following sections provide a detailed overview of these key integrated functionalities.

5.1.1 Projects Management

In order to keep the product backlog separate, GenSP allows users. Each project operates as an independent workspace and contains its own set of modules. For example, every project maintains a unique product backlog consisting of multiple backlog items, ensuring that work items from different projects remain organized and

do not overlap. The project management module supports all basic CRUD (Create, Read, Update, and Delete) operations, enabling users to easily add new projects, view project details, modify information, and remove projects as needed. This structure provides a clear and efficient starting point for managing Agile workflows and keeps project data organized throughout the development process.

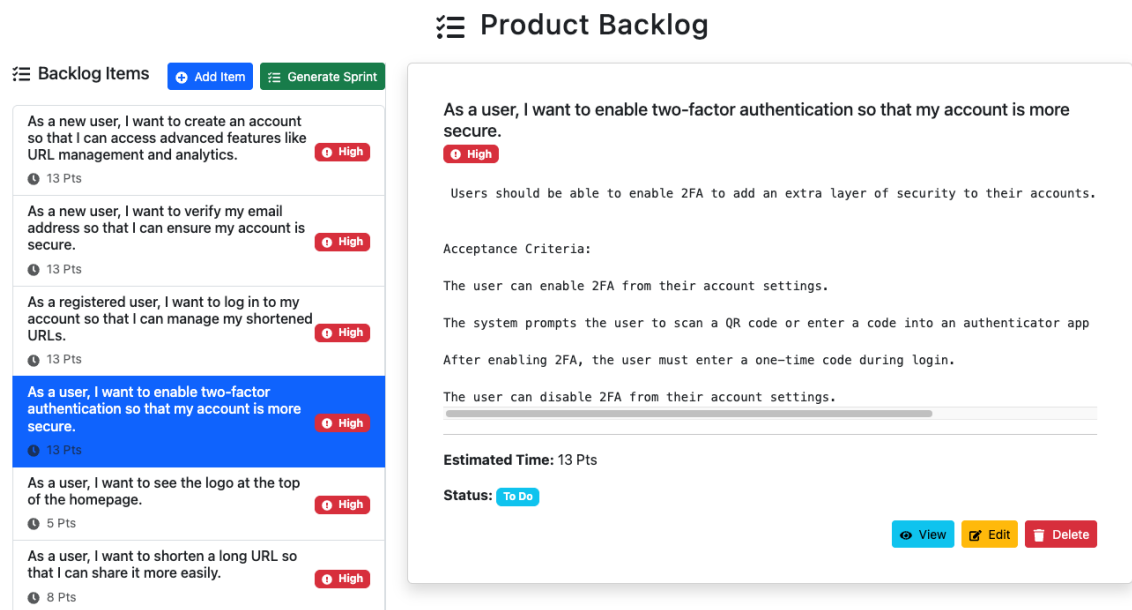


Figure 5.1: GenSP Product Backlog.

5.1.2 Backlog Management

The product backlog is a fundamental component of Scrum, and GenSP provides robust features for managing backlog items through comprehensive CRUD operations. In the current version, users can create and manage different types of backlog items, including epics, stories, bugs, and tasks. Each item can be assigned a priority level—High, Medium, or Low—and its status can be updated to reflect its progress, such as To Do, In Progress, or Done. This flexibility allows teams to adjust item statuses as work advances during a sprint. Additionally, user story points can be entered manually or automatically generated using GenAI, supporting more consistent

and accurate effort estimation throughout the development process.

5.1.3 Story Points Estimation

When a backlog item is created in GenSP, there is an option to generate user story points automatically. GenAI estimates the story points based on the item's title and description, including the acceptance criteria. Once generated, the estimated story points are entered directly into the appropriate field. Users can also click the "Details" button to view an explanation from GenAI, describing the reasoning behind the assigned story points. This feature provides a valuable starting point for team discussions, helping to save time and streamline the process of estimating user story points.

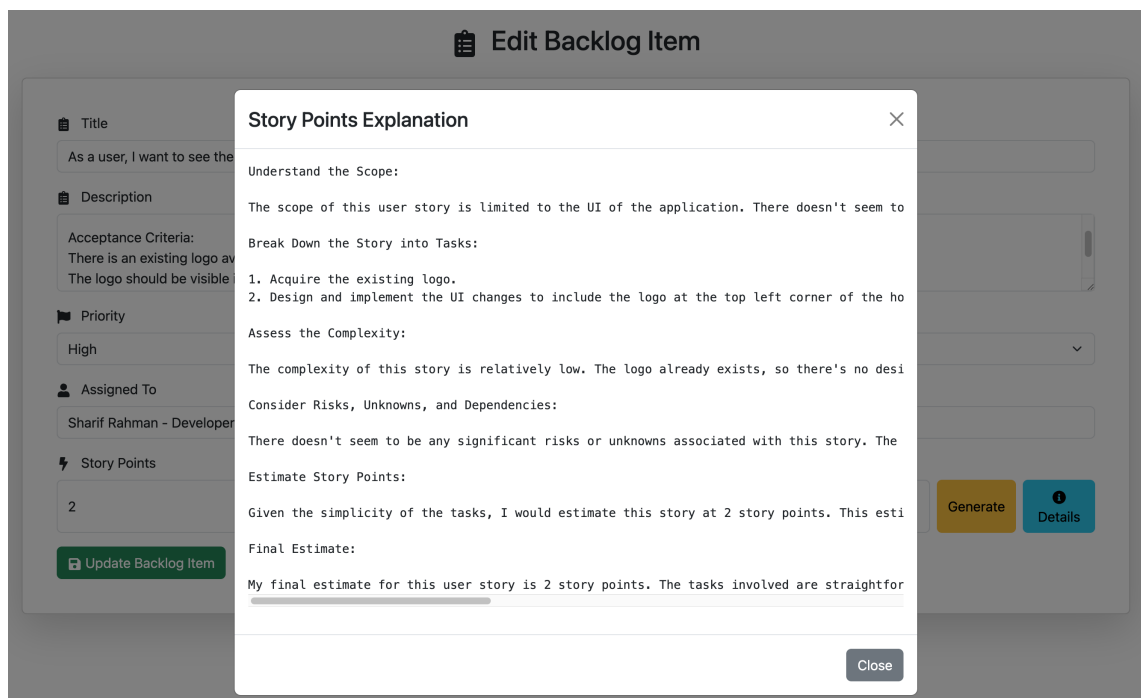


Figure 5.2: GenSP Story Point Generation.

5.1.4 Sprint Generation

GenSP enables GenAI-assisted sprint backlog generation to streamline the planning process. Users begin by entering the sprint goal and specifying the sprint duration in days (for example, entering “10” for a 10-day sprint). Next, the development team is selected, and each member’s total available working hours for the sprint are provided (e.g., 80 hours for a developer available all 10 days, assuming an 8-hour workday). After entering these details, the user can click the “Generate Sprint” button. GenSP will then utilize ChatGPT’s API to intelligently suggest sprint backlog items, taking into account unfinished backlog items, the defined sprint goal, and the team’s total capacity. This feature helps ensure that the generated sprint plan aligns closely with project objectives and team availability.

5.1.5 Task Breakdown

GenSP also provides automatic user story breakdown, helping users divide a user story into smaller, manageable tasks. When a user story point is generated, GenAI simultaneously analyzes the story’s title and description to suggest a set of subtasks. This breakdown not only clarifies the complexity and scope of work required to complete the user story but also helps identify potential dependencies that might otherwise be overlooked.

5.1.6 Task Assignment

GenSP also features intelligent task assignment during sprint backlog generation. After receiving the sprint goal and team capacity, GenSP uses ChatGPT’s API to recommend which backlog items to include and assigns them to the most suitable team members based on their roles (e.g., frontend developer, backend developer, UI designer) and their past tasks. While team preferences may still influence final assignments, GenSP provides a draft plan that streamlines planning and supports

balanced workload distribution, making it easier for teams to review and adjust during sprint planning.

5.1.7 User Authentication and Management

The system includes essential user management features, such as user authentication, to ensure security. For simplicity, only users with the admin role are permitted to manage user accounts through CRUD operations. All other users can log in using their email and designated password and may only modify their own information. The role-based access controls are kept straightforward, as advanced user permissions are not directly relevant to improving sprint planning. However, implementing basic authentication helps protect the system from unauthorized access while allowing legitimate users to securely use the application.

5.2 GenSP System Architecture

The GenSP application is designed with a modular architecture that integrates four key components: the frontend, backend, database, and GPT API integration. Figure 2.1 presents the high-level architecture of the GenSP application. This architecture ensures a seamless flow of information between users and the system, enabling efficient and AI-enhanced sprint planning. The frontend offers an intuitive user interface for managing all sprint planning activities, while the backend handles business logic, data processing, and communication with both the database and GPT API. The database maintains persistent and secure storage of all project data. Integration with the GPT API empowers the system to provide intelligent suggestions for user stories, estimations, and task breakdowns.

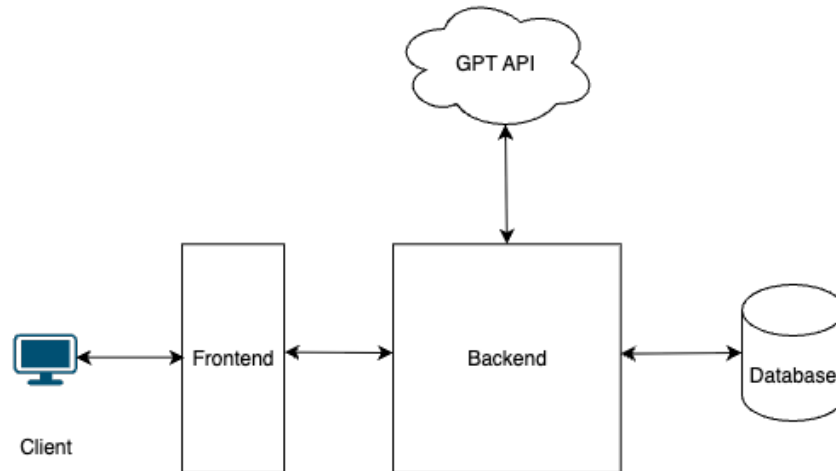


Figure 5.3: GenSP System Architecture.

5.2.1 Backend Implementation

The backend serves as the central logic and coordination layer of the GenSP application. Its primary responsibility is to process incoming requests from the frontend, execute core business logic, and manage seamless data exchange between the user interface, the database, and external services such as the ChatGPT API. The backend is structured around a set of RESTful endpoints, each designed to support specific operations required for effective sprint planning and project management.

5.2.2 Technologies and Tools

The backend of GenSP acts as the central logic layer, processing frontend requests, managing business logic, and coordinating data flow between the UI, database, and external services like the ChatGPT API. It is organized around RESTful endpoints tailored for sprint planning and project management.

Technologies and Tools

GenSP uses a modern, scalable stack:

- **Express.js:** This Node.js framework powers the RESTful API, supporting

scalable and robust backend logic [63].

- **MongoDB:** A flexible NoSQL database, well-suited for managing Agile project data [64].
- **Mongoose:** ODM library providing schema-based data modeling and validation for MongoDB [65].
- **OpenAI GPT API:** Integrated via OpenAI's Node.js SDK to enable GenAI-powered sprint planning features [66].

Project Structure

GenSP adopts a modular, MVC-inspired architecture (excluding the traditional view layer since the frontend consumes API endpoints). The service-repository pattern enhances code maintainability and scalability [67]. Models define schemas, controllers handle requests, services implement business logic, and repositories manage database interactions.

- **Route:** Exposes RESTful endpoints for project operations.
- **Middleware:** Validates requests and manages permissions.
- **Model:** Defines project schemas in MongoDB via Mongoose.
- **Controller:** Processes project-related HTTP requests.
- **Service:** Handles business logic for project tasks.
- **Repository:** Abstracts MongoDB interactions.

Database Design

MongoDB's document-based structure and Mongoose's schemas provide flexible and scalable data management, supporting clear relationships between entities like

projects, users, backlog items, sprints, and tasks. This approach ensures data integrity and efficient CRUD operations throughout the application.

Security and Best Practices

Security best practices include encrypted password storage, JWT-based authentication, and rigorous input validation. These measures protect user data and maintain system integrity.

5.2.3 ChatGPT's API Integration

A central feature of the GenSP application is its integration with OpenAI's ChatGPT API to enable intelligent automation for sprint planning tasks. The backend leverages LLM to generate user stories, estimate story points, break down tasks, and assign responsibilities. This chapter details how the ChatGPT API is integrated, the specific techniques used for prompt engineering (such as chain-of-thought prompting), how context is maintained across API calls, and the role of the Node.js OpenAI SDK.

The GenSP system's backend communicates with ChatGPT using the official OpenAI Node.js SDK. This SDK makes it easier to send requests to OpenAI's API and get responses back. This way, GenAI features can be added smoothly to REST endpoints. Important steps for this integration include setting up the system by securely storing API keys. All interactions with the OpenAI API are handled in a specific service module to keep things organized. The system also includes strong error handling to catch problems with the API and provide helpful information to the user if something goes wrong.

To begin, an OpenAI client must be created. This client acts as the connection point to the OpenAI API. Once the client is set up, it can be used to send requests to ChatGPT. This process typically involves specifying the model to use and providing

the user's message. ChatGPT then processes this request and sends back a response. The system receives this response from the API via the client.

```
// Load environment variables from .env file
dotenv.config();
const configuration = new Configuration({
  apiKey: process.env.OPENAI_API_KEY,
});
const openai = new OpenAIApi(configuration);

export default openai;
```

Before sending a request to the ChatGPT API, the system first selects the appropriate model—"gpt-4" in this case—due to its advanced intelligence. The request is then prepared by constructing a sequence of messages. This includes a system message that sets the AI's role, such as "You are an Agile Sprint Planner." Finally, the main prompt is tailored to the specific sprint planning task at hand, whether it involves backlog generation, story point estimation, or task breakdown. A temperature parameter, often set to 0.5, is specified to balance creativity and predictability in the AI's responses. Once the messages and settings are assembled, the backend sends the request to the OpenAI chat completions API [68] and waits for the AI-generated response, which is then processed and integrated back into the application workflow.

```
const response = await openai.chat.completions.create({
  model: "gpt-4",
  messages: [{
    role: "assistant",
```

```
        content: "You are an Agile Sprint Planner."
    },
    {role: "user", content: prompt}
  ],
  temperature: 0.5,
});
```

5.2.4 Frontend Implementation

Initially, the scope of this project did not include a frontend implementation. However, it soon became clear that developing a frontend interface was crucial. The frontend helps users clearly visualize how GenAI can enhance Agile sprint planning by making it interactive and user-friendly. For this purpose, React.js, one of the most popular and robust frontend frameworks, was chosen for development.

The frontend architecture of GenSP follows a modular structure, mirroring the organization of backend modules such as Projects, Backlog Items, Sprint Generation, and User Management. Each frontend component corresponds to its respective backend module, which ensures a consistent, clear, and maintainable codebase. This modular design allows components to be developed, tested, and maintained independently, facilitating efficient collaboration and scalability. Furthermore, aligning frontend modules with backend logic simplifies data handling and enhances the overall coherence of the application, providing a seamless user experience.

For the GenSP frontend implementation, two key libraries were used: Axios and Bootstrap. Axios was chosen as the primary HTTP client due to its simplicity, efficient handling of API requests, built-in promise support, and clear error management. It streamlined communication between the React frontend and the Express.js backend by making asynchronous requests easy and maintainable. Bootstrap was integrated to provide rapid development of the application's styling and responsive

layout. Its robust, grid-based system and pre-built components enabled quick creation of an intuitive and visually appealing user interface, ensuring compatibility across various devices and screen sizes.

5.3 Prompt Engineering

A critical component of integrating GenAI into the GenSP application is designing effective prompts for the ChatGPT API. Prompt engineering is the practice of designing and structuring prompts to effectively communicate tasks to AI models [69]. The quality and structure of these prompts directly influence the relevance and usefulness of the AI-generated outputs. Its benefit is that well-crafted prompts improve the relevance, accuracy, and fairness of AI outputs while enabling the model to handle complex problems and reducing errors [69].

For each sprint planning task, prompts are carefully crafted to provide clear instructions and guide the LLM through a logical reasoning process. CoT technique is used where prompts include not only the question but also explicit, step-by-step reasoning (rationales) leading to the answer, encouraging the model to show its intermediate thought process before giving the final response [16].

For example, when estimating user story points, the prompt is constructed to simulate the thought process of an experienced Agile team member. The prompt instructs the AI to analyze the user story by first identifying its key components—such as user interface, backend, database, or API requirements. The AI is then guided to break down the story into smaller subtasks, assess the complexity of each subtask, and consider factors such as technical difficulty, dependencies, and potential risks.

To encourage thorough and systematic reasoning, the prompt uses a step-by-step chain-of-thought approach. This includes explicit steps for understanding the story's scope, evaluating unknowns, and estimating the required effort using the Fibonacci sequence—a standard practice in Agile estimation. The prompt also directs the AI

to provide both a numeric estimate and a summary of the reasoning behind the suggested story points, ensuring that the output is transparent and actionable for the development team.

By employing detailed, task-specific prompts like the example below, GenSP maximizes the value of GenAI in sprint planning. This structured prompt-building process helps produce consistent, logical, and context-aware recommendations that align with real-world Agile practices.

```
const prompt = `
Act as an Agile team member responsible for estimating story points
for a user story. Follow this step-by-step chain of thought to
estimate the story points based on the provided story title,
description, and acceptance criteria. Be detailed and logical
in your reasoning.

Understand the Scope:
Identify the key components of the story, such as UI, Backend,
Database, API, etc.

Break Down the Story into Tasks:
List the specific subtasks required to complete the story.

Assess the Complexity:
Evaluate the complexity of each subtask. Consider factors like
technical difficulty, dependencies, and effort required.

Consider Risks, Unknowns, and Dependencies:
Identify any potential risks, unknowns, or dependencies that
```

could impact the effort required.

Estimate Story Points:

Based on the above analysis, suggest a story point estimate using the Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.).

Assume that 1 story point represents a day of work for an experienced team member and all the members of the development team is highly experienced and familiar with the project.

Final Estimate:

Provide the final story point estimate and a brief summary of the reasoning. Use floor or ceiling numbers for the total story points estimate so that it follows Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.).

Here is the user story title, description, and acceptance criteria:

Title: \${storyTitle}

Description: \${storyDescription}

Now, follow the chain of thought and provide a detailed story point estimate with proper reasoning.`;

To ensure ChatGPT provides relevant and connected responses, especially during ongoing planning, the GenSP backend carefully builds a messages array for each API

request and stores it as chat history. For tasks that involve multiple exchanges, like generating a sprint backlog, the conversation history is also included. This helps the AI remember previous responses and user follow-ups, allowing it to learn from how past sprints were managed, including story selection, effort estimations, task breakdowns, and team assignments.

5.4 Challenges and Lessons Learned

The development of the GenSP application presented various technical and design challenges that shaped the final outcome of the project. One of the most significant challenges was the evolving scope of the system. Initially, the goal was to develop a lightweight API focused solely on sprint planning and backlog management. However, as development progressed and real-world use cases were considered, it became evident that broader project management capabilities were essential. This shift required the introduction of new features, such as multi-project support, team member assignments, and user role management. As a result, the backend needed substantial restructuring to accommodate these changes, including updates to data models, API endpoints, and access control logic.

Integrating the GPT API posed another key challenge. Crafting prompts that produced accurate and consistent results required multiple iterations. For tasks like sprint backlog generation or story point estimation, prompts had to be specific enough to guide the AI's reasoning but concise enough to remain within token limits. To maintain output reliability, a two-step prompt structure was adopted—first to generate an explanation, then to request structured output. However, tuning this interaction was not straightforward. A common difficulty involved balancing the creativity and consistency of the AI's responses, as large language models like GPT are prone to variability.

Another observed issue was hallucination, where the AI produced plausible but

incorrect or exaggerated outputs. For example, when estimating story points, the AI occasionally assigned high estimates if the story lacked a clear description or defined acceptance criteria. This highlighted the importance of prompt clarity and well-prepared input data.

From a technical standpoint, ensuring smooth communication between the backend and frontend was also a challenge. The dynamic nature of AI-generated suggestions meant that API responses needed to be handled carefully, with proper state management and real-time UI updates. Security was another important concern. Protecting sensitive data, managing user access, and securing API keys required dedicated attention throughout the implementation.

Several important lessons were learned during this process. First, project requirements can change quickly, so it is important to design flexible systems that can adapt over time. Successful integration of GenAI depends heavily on precise prompt engineering and strong error handling. Incorporating AI into a traditional project management system is not trivial—it involves managing dependencies between various modules while maintaining the logic and usability of the system. Involving users early in testing can reveal valuable insights and usability issues that guide further development. Finally, maintaining a clear separation of concerns across the backend, AI logic, and frontend layers proved vital for scalability and long-term maintainability.

Overall, these experiences underscored the importance of iterative development, adaptability, and collaboration. Building an AI-assisted tool like GenSP required not only technical skill but also a mindset open to learning and continuous refinement.

6 Results and Discussion

The evaluation in this thesis was conducted using two methods. First, the GenSP tool was tested against completed sprint data from the MyFlavoria project backlog to ensure its key features were assessed with realistic project information. Second, a structured survey was carried out among experienced Agile practitioners and software developers.

6.1 Evaluation using MyFlavoria Project

MyFlavoria [60] is an app that lets customers track their meals, view nutritional intake, and monitor biowaste at the Flavoria research restaurant, contributing to food innovation and waste reduction research. To assess the effectiveness of GenSP's AI-assisted sprint planning, the tool was applied to a real-world scenario in the MyFlavoria software project. Its product backlog, historical sprint data, and previous sprint goals provided a solid benchmark for comparison. The evaluation focuses on how GenSP performed in user story breakdown, story point estimation, task assignment, and overall sprint planning, relative to the project's traditional sprint history.

For this project, the team consisted of seven members: four software engineers and three UI/UX designers. The software engineers formed the development team, while the UI/UX designers made up the design team. The members are referred to as follows:

- Software Engineer 1 (SE1)
- Software Engineer 2 (SE2)
- Software Engineer 3 (SE3)
- Software Engineer 4 (SE4)
- UI/UX Designers 1 (UX1)
- UI/UX Designers 2 (UX2)
- UI/UX Designers 3 (UX3)

6.1.1 User Story Breakdown and Estimation

GenSP was used to analyze user stories from the MyFlavoria backlog. For each user story, the tool generated a breakdown of actionable tasks and provided story point estimates. Table 6.1 illustrates this process by mapping representative user stories to their respective GenSP-generated tasks and point estimates:

Table 6.1: User Stories, Subtasks, and Estimated Points

User Story	Broken Down Tasks	Story Points
As a student, I want to see how my lunch compares to national dietary recommendations, so that I can adjust my diet to make it healthier.	<ul style="list-style-type: none"> - Design comparison UI - fetch dietary API - Integrate with user data - Testing 	8

As a diner, I want to see today's menu so I can decide what I want to eat	<ul style="list-style-type: none"> - Design today's menu UI - Fetch menu API - Integrate frontend - Test menu display 	3
As a diner/allergic I want to be able to inspect the ingredients of each food option (in the restaurant and elsewhere) so I know what I want to eat	<ul style="list-style-type: none"> - Add ingredient info to DB - Ingredient info UI - Modal/pop-up - Test feature 	5
As a diner I want to see the week's menu in the restaurant so I know what they're serving on other days	<ul style="list-style-type: none"> - Fetch weekly menu - Week-view menu UI - Integrate with frontend - Testing 	3
As a vegetarian/vegan, I want to see the vegan food option first in the menu so I don't have to go through options not meant for me, and as a meat eater, I want to see the meat food options first in the menu so I don't have to go through options not meant for me	<ul style="list-style-type: none"> - Dietary preference settings - Sorting/filter logic - UI for ordering - Test personalization 	5

<p>As a diner I want to very easily be able to view the QR code needed at checkout so that it doesn't slow my checkout</p>	<ul style="list-style-type: none"> - QR access button UI - Integrate QR generator - Test QR display 	3
<p>As a diner, I want the garbage point to be less crowded so I can exit the restaurant fast</p>	<ul style="list-style-type: none"> - Analyze current process - Design solution - Implement notifications/alerts - Test with users 	8
<p>As a diner, I want to know when the QR code has been successfully read at the checkout so that I do not have to guess if it has been properly read or not.</p>	<ul style="list-style-type: none"> - Feedback UI for scan - Integrate with checkout - Test response 	2
<p>As a restaurant owner I want to inform possible customers about the opening hours and location of the restaurant, so that customers will find their way to the restaurant and at the correct time.</p>	<ul style="list-style-type: none"> - Manage info API fetch - Design info UI - Test visibility 	2

As a diner I want to be able to find the restaurant open times easily so I know when I can eat there	<ul style="list-style-type: none"> - Quick access UI - Backend integration - Test feature 	2
As a restaurant employee responsible for the food I want to get as much customer feedback as possible so I can make changes to the menu	<ul style="list-style-type: none"> - Feedback UI - Feedback storage through API - Admin analytics view - Test feedback process 	5
As a new user I want a video version of instructions so I can easily understand how the app and the restaurant works	<ul style="list-style-type: none"> - Script and record video - Integrate in onboarding - Test accessibility 	3
As a non-finnish-speaking customer, I want to be able to change the language of the application so I can understand it	<ul style="list-style-type: none"> - Language selection UI - Translate content - Dynamic switching - Test functionality 	8

This approach demonstrated that GenSP could produce detailed, actionable task breakdowns and realistic effort estimations, comparable to what experienced Scrum teams deliver manually.

6.1.2 Sprint Backlog Generation and Task Distribution

Beyond analyzing individual user stories, GenSP generated complete sprint plans based on sprint goals, team capacity, and backlog status. It selected user stories, defined sprint goals, and recommended balanced task assignments. These plans were then compared to the MyFlavoria project's historical sprints.

Table 6.2: GenSP Suggested Sprint and Task Distribution

Sprint	Sprint Goal	Selected User Stories	Assigned Workers
1	Launch core menu and essential info	2 (Today's menu), 4 (Week's menu), 9 (Owner info), 10 (Open times)	SE1: Today's menu API, SE2: Week's menu API, SE3: Integration, SE4: Owner/Open times backend; UX1: Today's menu UI, UX2: Week's menu UI, UX3: Owner/Open times UI
2	Personalize menu and enable QR basics	5 (Menu sorting), 6 (QR access), 8 (QR scan feedback)	SE1: Sorting logic, SE2: QR integration, SE3: Profile settings, SE4: Scan feedback logic; UX1: Sorting UI, UX2: QR UI, UX3: Scan feedback UI

Continued on next page

Table 6.2 – continued from previous page

Sprint	Sprint Goal	Selected User Stories	Assigned Workers
3	Dietary safety and language preparation	3 (Ingredients inspection), 13 (Language change - backend & UI setup)	SE1: Ingredients backend, SE2: Language selection backend, SE3: Ingredient modal, SE4: Content translation; UX1: Ingredients UI, UX2: Language UI, UX3: Testing
4	Dietary comparison & feedback collection	1 (Dietary comparison), 11 (Feedback collection)	SE1: Dietary backend, SE2: Feedback backend, SE3: Integration, SE4: Feedback analytics; UX1: Comparison UI, UX2: Feedback UI, UX3: Feedback admin UI
5	Onboarding, video, and garbage solution	7 (Garbage point), 12 (Video instructions)	SE1: Garbage backend logic, SE2: Video integration, SE3: Garbage solution UI, SE4: Testing; UX1: Video content, UX2: Garbage UI, UX3: Onboarding UI

Continued on next page

Table 6.2 – continued from previous page

Sprint	Sprint Goal	Selected User Stories	Assigned Workers
6	Refine menu features and multi-lingual UI	2 (Menu refinements), 4 (Week’s menu refinements), 13 (Language finalization)	SE1: Menu logic refinements, SE2: Language switching, SE3: Menu UI testing, SE4: Language testing; UX1: Menu UI polish, UX2: Language UI review, UX3: Testing
7	User feedback & polish	1 (Dietary comparison polish), 11 (Feedback improvements), Bug fixes	SE1: Dietary improvements, SE2: Feedback UI/UX, SE3: Bug fixing, SE4: Testing; UX1: UI feedback, UX2: Feedback UI, UX3: Testing
8	Final review, documentation, re-release	All stories (final testing, docs, re-release prep)	SE1: Final testing, SE2: Documentation, SE3: Release prep, SE4: Deployment; UX1: User guide, UX2: Final UI polish, UX3: Accessibility review

6.1.3 Overall analysis

A detailed analysis is conducted by manually reviewing the historical sprint documentation of the MyFlavoria project and comparing it to the sprint plans generated by GenSP. The comparison focused on specific sprint planning criteria, such as completeness of backlog items, alignment between sprint goals and backlog, workload distribution among team members, and identification of task dependencies. The

objective is to evaluate how effectively GenSP addresses identified challenges in traditional sprint planning.

- **Challenge: Incomplete Backlog Items**

Backlog items are sometimes added without clear descriptions or well-defined acceptance criteria. This lack of detail can cause confusion and lead to misunderstandings during development and testing.

GenSP Approach: GenSP can break down high-level backlog items into smaller tasks and suggest relevant details. This provides a starting point for the team to clarify requirements and acceptance criteria, leading to better understanding and smoother development.

- **Challenge: Misalignment Between Sprint Goals and Backlog**

In the initial sprint, the backlog often does not reflect the actual sprint goals. Both the goals and the user stories may be unclear, and their connection to each other is often weak. This misalignment can make it difficult for the team to focus and measure progress.

GenSP Approach: GenSP helps by suggesting backlog items that are closely linked to the stated sprint goals. This ensures that each sprint has a clear purpose and that the selected user stories directly contribute to achieving the sprint objectives. It also encourages meaningful team discussions about priorities.

- **Challenge: Unbalanced Workload Distribution**

In some sprints, senior team members are overloaded with work while others are underutilized. This creates bottlenecks and can slow down the team's overall progress.

GenSP Approach: GenSP considers each team member's skills and current workload when recommending task assignments. This results in a more

balanced distribution of work, reduces bottlenecks, and allows everyone to contribute according to their strengths.

- **Challenge: Overlooked Task Dependencies**

Dependencies between tasks can be missed during planning, which may cause delays or require rework later in the sprint. For example, a user interface cannot be implemented before its design is ready.

GenSP Approach: GenSP automatically identifies and highlights potential dependencies during the task breakdown phase. This allows the team to address dependencies early, reducing the risk of delays and rework during the sprint.

Disadvantages

Despite its benefits, the GenSP tool also demonstrated certain limitations during the analysis. Firstly, GenSP occasionally generated unnecessary subtasks that were not relevant to completing the user story. For example, for the story *"As a diner/allergic, I want to be able to inspect the ingredients of each food option. . . ,"* the tool suggested a subtask to *"Add ingredient info to DB"*. However, in the context of the project, the ingredient information was already managed by an existing API, making this subtask redundant. This issue could be mitigated by providing clearer user story descriptions and more precise acceptance criteria.

In a few instances, GenSP appeared to overestimate story points, potentially leading to inaccurate workload planning. This, too, highlights the importance of well-defined user stories to support more accurate estimations.

The tool sometimes assigned testing tasks to multiple developers within the same sprint, when it would have been more efficient to delegate such tasks to a single developer, freeing others to focus on additional development work.

The evaluation with the MyFlavoria project shows that GenSP improves Agile

sprint planning by refining backlogs, aligning backlog items with sprint goals, providing effort estimates, and balancing task assignments. While GenSP addresses several challenges of traditional methods and enhances planning clarity, it still has some shortcomings that require human oversight. Overall, integrating GenAI into sprint planning can lead to more effective and predictable project outcomes.

6.2 Survey Results

In this section, the survey results with participants' demographics, quantitative as well as qualitative results, and implications are discussed. The complete set of survey questions is provided in the appendix section.

6.2.1 Survey Demographics

A total of nine software engineers and developers voluntarily participated in this survey. All participants currently hold positions as software engineers or developers (100%), ensuring that the feedback reflects the perspective of professionals actively engaged in software development.

Table 6.3: Participant's current company location

Company Location (Country)	Participant Count	Percentage
Finland	3	33.3%
United States	2	22.2%
Bangladesh	2	22.2%
Sweden	1	11.1%
United Kingdom	1	11.1%

The respondents represent a diverse set of countries and work environments. As shown in Table 6.3, participants are employed in Finland (33.3%), the United States (22.2%), Bangladesh (22.2%), Sweden (11.1%), and the United Kingdom (11.1%). This international mix brings a range of experiences and viewpoints from Scrum teams operating in different organizational and cultural settings.

The respondents are highly experienced in software development, with the majority occupying senior roles. Table 6.4 details their years of experience:

Table 6.4: Participant’s software development experience

Experience(Years)	Participant Count	Percentage
3–5 years	1	11.1%
6–10 years	5	55.6%
More than 10 years	3	33.3%

This shows that almost 90% of the participants have more than six years of industry experience, ensuring that their insights are grounded in extensive professional practice.

All respondents reported familiarity with the Scrum framework. Most (88.9%) identified themselves as having intermediate experience, typically as regular Scrum team members, while one respondent (11.1%) indicated advanced knowledge, including experience as a Scrum Master or Product Owner.

Table 6.5: Familiarity with Scrum

Company Location (Country)	Participant Count	Percentage
Intermediate (Scrum team member)	8	88.9%
Advanced (Scrum Master/Product Owner)	1	11.1%

Additionally, all participants (100%) reported using project management tools such as Jira, Trello, or similar software, indicating familiarity with common Agile project practices.

All survey participants reported familiarity with GenAI tools, including technologies such as ChatGPT, Gemini, or GitHub Copilot. However, 55.6% indicated that they had not previously used GenAI specifically for sprint planning, while 44.4% reported using such tools for sprint planning only occasionally or on a trial basis.

As presented in Table 6.4, the participant group includes software professionals with experience in Scrum, project management, and exposure to GenAI technologies. These characteristics provide important context for interpreting their perspectives

on the integration of GenAI into Scrum sprint planning. Prior to completing the survey, participants were shown a live demonstration or video walkthrough of the GenSP tool to ensure they understood how GenAI can be applied in sprint planning.

6.2.2 Quantitative Results

This section presents the analysis of quantitative data collected through Likert-scale survey questions. The results are organized according to key use cases of GenAI in sprint planning, including backlog creation, story point estimation, task breakdown, and task assignment. Each use case is evaluated based on participants' responses, providing measurable insights into the perceived value, usefulness, and potential challenges of integrating GenAI into the sprint planning process.

GenAI in Sprint Backlog Generation

The results suggest that GenAI is viewed positively for supporting sprint backlog creation. As shown in Table 6.6, 77.8% of respondents rated GenAI as valuable for creating the sprint backlog, citing its ability to suggest user stories and potentially reduce human error and bias. No respondents expressed a negative view, while 22.2

Additionally, the survey asked about GenAI's ability to identify missing details, dependencies, or acceptance criteria for items proposed for the sprint backlog. Here, 66.7% of participants saw GenAI as valuable, though a small proportion (22.2%) expressed skepticism. This finding suggests that while most practitioners appreciate the AI's potential for refining backlog items and flagging gaps, some may be cautious about relying on AI for complex, context-dependent decisions.

When it comes to prioritizing or ordering items within the sprint backlog, responses were more mixed: 44.4% saw GenAI as helpful, 44.4% were neutral, and 11.1% were negative. This result implies that, while GenAI is appreciated for its automation and analytical power, practitioners still rely on their experience and

Table 6.6: Survey Results on GenAI in Sprint Backlog Generation

Survey Question	Positive	Neutral	Negative
<i>How valuable do you think generative AI could be in helping to create the sprint backlog (i.e., suggesting user stories based on sprint goal, capacity, and previous sprint information, reducing human error and bias)?</i>	77.8%	22.2%	0%
<i>How valuable could generative AI be in identifying missing details, dependencies, or acceptance criteria for items proposed for the sprint backlog?</i>	66.7%	11.1%	22.2%
<i>How helpful could generative AI be in suggesting prioritization or ordering of items within the sprint backlog?</i>	44.4%	44.4%	11.1%

understanding of project priorities for final decision-making.

Further insights were gathered through this question: *“In your opinion, during sprint backlog creation, which area could benefit the most from the use of generative AI?”* the most frequent responses were backlog item refinement improving descriptions and acceptance criteria (77.8%), sprint backlog suggestions based on goal and capacity (66.7%), and reprioritization of backlog items (33.3%). These results reinforce the view that GenAI is especially well-suited to tasks involving analysis and refinement of backlog content, as well as making context-aware suggestions.

However, participants also highlighted key challenges and limitations of using GenAI in this context. The most common concern was understanding complex business logic and dependencies (9 votes). This reflects that GenAI may not yet fully capture the deeper nuances and cross-functional dependencies that experienced team members can identify.

In summary, the survey results show that experienced Scrum practitioners see significant value in using GenAI for sprint backlog generation, particularly for backlog refinement and suggesting backlog items aligned with sprint goals and capacity. However, they also recognize the current limitations of AI in understanding complex business logic and dependencies.

GenAI in Story Points Estimation

The survey investigated participants' perceptions of using GenAI for estimating story points during sprint planning. The results reveal mixed attitudes toward the reliability and usefulness of AI-generated estimates. When asked whether they agree that *"Generative AI can estimate user story points based on story title, descriptions, and historical data,"* only 33.3% responded positively, while 22.2% were neutral, and a notable 44.4% expressed disagreement. This indicates that, although some practitioners see value in AI-powered estimation, many remain cautious about fully trusting these automated outputs.

However, when it comes to the collaborative aspect of estimation, the responses were more favorable. Over half of the participants (55.6%) agreed that *"AI-proposed estimates or insights help our team reach consensus on story point estimations more quickly,"* suggesting that GenAI can serve as a useful facilitator in team discussions. Similarly, 55.6% said they would be comfortable using an AI-generated estimate as a starting point for team discussion, rather than relying solely on traditional methods like Planning Poker. Still, one-third of respondents expressed hesitation or discomfort with this approach.

Table 6.7: Survey Results on GenAI in Story Points Estimation

Survey Question	Positive	Neutral	Negative
<i>"Generative AI can estimate user story points based on story title, descriptions and historical data." How much do you agree with this statement?</i>	33.3%	22.2%	44.4%
<i>"AI-proposed estimates or insights help our team reach consensus on story point estimations more quickly." – Please rate your agreement.</i>	55.6%	11.1%	33.3%
<i>Would you be comfortable using an AI-generated estimate as a starting point for team discussion? (e.g., instead of Planning Poker)?</i>	55.6%	11.1%	33.3%

Participants also identified key challenges in leveraging GenAI for story point estimation. The most frequently mentioned concern was the AI's ability to handle novel or complex user stories that require a deep contextual understanding (8 votes). This feedback underscores the current limitations of GenAI in dealing with the nuances and complexities that arise in real-world software projects.

Overall, fewer participants are convinced that GenAI can independently provide accurate story point estimates. However, GenAI can be valuable to speed up team consensus and as a starting point for discussions.

GenAI in Story/Task Breakdown

The survey explored participants' perceptions of using GenAI to break down user stories into tasks or subtasks, as well as its ability to identify dependencies between those tasks. The findings show that a majority of respondents (66.7%) agree that GenAI is helpful in creating clear lists of tasks or subtasks from user stories. This indicates a positive attitude toward leveraging AI to support this traditionally manual and sometimes inconsistent aspect of sprint planning. Only 11.1% of respondents expressed a negative view, suggesting that most software professionals are open to at least experimenting with AI-assisted breakdowns.

When asked about GenAI's usefulness in identifying potential task dependencies—either within a single user story or across multiple stories—over half of the participants (55.6%) viewed this as a helpful feature, while 44.4% remained neutral, and no participants expressed a negative view. This result highlights that, while some teams are optimistic about AI's ability to surface hidden or complex dependencies, others may still be evaluating its reliability and accuracy in real-world scenarios.

The most common concern, cited by eight participants, was the challenge of *“understanding the specific technical implementation approach.”* This underscores

Table 6.8: Survey Results on GenAI in Story/Task Breakdown

Survey Question	Positive	Neutral	Negative
<i>“Generative AI is helpful in breaking down user stories into a clear list of tasks or sub-tasks.” – How much do you agree?</i>	66.7%	22.1%	11.1%
<i>How helpful could generative AI be in identifying potential task dependencies (within the story or to other stories)?</i>	55.6%	44.4%	0%

a critical limitation: while GenAI can effectively suggest initial breakdowns and surface potential dependencies, it may not always capture the complex technical details or context-specific solutions that teams require for successful delivery.

In summary, the survey results indicate that participants see potential for GenAI to assist with story and task breakdowns, as well as identifying dependencies.

GenAI in Task Distribution

The survey also examined how practitioners view the use of GenAI in breaking down user stories into tasks and distributing those tasks among team members. The responses indicate a generally positive outlook, with 66.7% of participants rating GenAI as helpful in suggesting task assignments based on individual skills, experience, or development goals. This suggests that most respondents believe GenAI can add value by leveraging available data to make better assignment decisions.

In addition, a majority of participants (55.6%) believe that GenAI could be valuable in balancing workload across team members during sprint planning. This ability to recommend fair and efficient task distribution is seen as a potential benefit, especially for larger teams or projects where manual balancing can be time-consuming and prone to oversight. However, 33.3% were neutral, and 11.1% viewed this application less favorably, reflecting that there are still reservations about the technology’s readiness and real-world effectiveness.

Participants were also asked to identify the main challenges they see in using

Table 6.9: Survey Results on GenAI in Task Distribution

Survey Question	Positive	Neutral	Negative
<i>How helpful could generative AI be in suggesting task assignments based on individual skills, experience, or development goals?</i>	66.7%	11.1%	22.2%
<i>How valuable could generative AI be in helping to balance the workload across team members during sprint planning?</i>	55.6%	33.3%	11.1%

GenAI for task distribution. The top concerns, each cited by six respondents, were accurately assessing individual skills and handling team preferences in task distribution by GenAI. These concerns highlight the complex, human-centric nature of task assignment.

In summary, the results suggest that software professionals are optimistic about the potential of GenAI to support task breakdown and assignment, particularly as a tool to enhance fairness and efficiency.

Overall Effectiveness of GenAI in Sprint Planning

The survey asked participants whether they believed GenAI could improve the overall effectiveness of their sprint planning process. The results show that a majority—55.6%—agreed that GenAI offers significant value, while 22.2% were neutral and 22.2% disagreed. This distribution suggests that most experienced software developers and Scrum practitioners see clear potential in GenAI.

To better understand where GenAI adds the most value, respondents were also asked which aspects of sprint planning benefit most from AI support (Figure 6.1). The most commonly selected area was improving planning efficiency, which is cited by seven participants. This aspect highlights GenAI’s ability to save time and reduce manual effort in preparing for sprints. User story refinement was the next most frequent response (six votes), which indicates that many teams recognize the usefulness of AI in clarifying requirements and acceptance criteria. Sprint backlog

creation and task breakdown/planning each received four votes, suggesting that GenAI is also seen as a valuable assistant in building and structuring the work for each sprint.

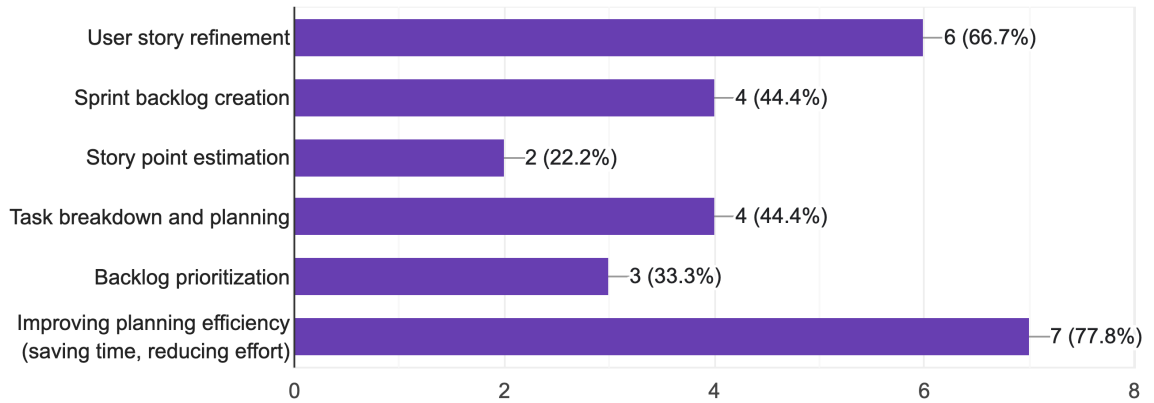


Figure 6.1: GenAI Benefits in Sprint Planning (Survey Results)

Despite some reservations, the survey results indicate that many participants recognize clear value in GenAI tools. Most participants agree that GenAI can positively impact sprint planning, especially for initial backlog creation and task breakdown. Respondents view GenAI not only as a means of automating routine tasks but also as a facilitator that can support more thoughtful and effective planning discussions. These findings suggest that GenAI tools have the potential to enhance both the efficiency and quality of sprint planning overall.

6.2.3 Qualitative Results

To complement the quantitative analysis, participants were invited to share their open-ended thoughts on the use of GenAI in sprint planning in the survey. The collected responses provide nuanced perspectives on the potential benefits, current limitations, and considerations for the adoption of GenAI in Agile processes.

Benefits and Opportunities

Several respondents highlighted clear advantages in using GenAI as a supportive tool during sprint planning. Participants noted that GenAI could significantly save time by automating routine tasks and providing smart suggestions for user stories, task breakdown, and estimation. One developer remarked,

"Using GenAI in sprint planning saves time, gives smart suggestions, improves task clarity, and helps teams plan better together."

One participant noted that GenAI can be very helpful in sprint planning, particularly for drafting user stories, distributing tasks among team members, and organizing work.

"I think using generative AI in sprint planning can be really helpful, especially for things like drafting user stories, helping distribute tasks among team members, and organizing tasks. It saves time and can surface useful suggestions or insights we might overlook."

This statement was echoed by others, who saw GenAI as a means to generate draft user stories, identify dependencies, and offer task distribution suggestions, thus streamlining the planning process. The following two statements from participants highlight the benefits and opportunities of using GenAI in sprint planning.

"It can be a purposeful tool for prioritizing tasks."

"I didn't used GenAI for sprint planning yet. But I think GenAI will be great tool to write use stories and task break down. If privacy ensured then I think GenAI can be helpful to do task break down, distribution and estimation by feeding previos sprints data."

One participant suggested that organizations could first try using GenAI tools with small teams or on smaller projects. This approach would allow them to see how well the tools work before using them on larger projects.

"Good idea. Small team and project can do trial and see results how effective it was and based on large scale projects can try to adapt it."

Concerns and Limitations

While many participants recognized the potential of GenAI in sprint planning, some expressed concerns about privacy, security, and the sensitivity of data. These respondents felt that the use of GenAI should be carefully controlled to protect sensitive information. For example, one participant remarked:

"I think we could use GenAI in a controlled manner considering security and sensitive information that should not be exposed. Additionally, we need to improve GenAI models to make them more accurate in the future."

A few participants also noted that current GenAI models may not yet be mature enough for reliable use in sprint planning. They pointed out that these tools are not yet able to understand or address the finer details that are often involved in the planning process. As one respondent stated:

"GenAI hasn't reached to a level of assessing the fine details required in sprint planning."

Despite these limitations, there was optimism about the future of GenAI in this area. Some participants expressed hope that ongoing improvements would eventually make GenAI a valuable resource for sprint planning and task distribution. As one participant commented:

"Still a long way to go for generative AI to be used in professional sprint planning. But I am sure some day AI will help tremendously in sprint planning and task distribution."

Balancing GenAI with Human Judgment

While recognizing the promise of GenAI, multiple respondents emphasize that human judgment and team context remain essential for effective sprint planning. As one participant states:

"I don't think it should replace human judgment since every team has its own way of working, and AI doesn't always understand the full context. I see it more as a supportive tool that helps streamline the process, not make the decisions for us."

Overall, the qualitative feedback indicates that experienced software practitioners view GenAI as a promising assistant for sprint planning—capable of improving efficiency, generating useful suggestions, and supporting the planning process. However, its current limitations, the irreplaceable value of human expertise, and important concerns around privacy and context highlight the need for careful, incremental adoption and ongoing improvements in GenAI technologies.

6.3 Mapping Research Questions with Results

The research questions guide both the practical evaluation of the GenSP tool and the structured survey of Agile practitioners. By mapping the research questions to the outcomes of these two approaches, the effectiveness and limitations of GenAI in sprint planning can be thoroughly assessed.

RQ1 addresses the value and effectiveness of GenAI in sprint backlog creation; both GenSP's practical outputs and survey responses provide evidence of its bene-

fits. GenSP was able to suggest relevant user stories that aligned well with sprint goals and helped identify missing details or dependencies that are often overlooked during manual planning. Survey participants also acknowledged that GenAI support in backlog creation improved the refinement process and reduced human error, although some noted challenges in handling complex business logic.

RQ2 focuses on GenAI's performance in user story point estimation. The GenSP tool demonstrated that GenAI could generate story point estimates with clear rationale, which helped facilitate team discussion and reduce subjective bias. Survey results showed a split in confidence, with some participants finding AI-generated estimates useful as a starting point, while others expressed concerns about the AI's ability to handle novel or complex stories. Nevertheless, the majority agreed that GenAI estimates could help teams reach consensus faster compared to purely manual methods.

RQ3 explores how GenAI assists in breaking down user stories into actionable tasks and identifying dependencies. GenSP's task breakdowns improve clarity and help teams recognize dependencies early. Survey respondents agreed that GenAI could generate detailed task lists and highlight potential dependencies, although there was some hesitation about relying solely on AI for technical implementation details. Many saw AI-driven breakdowns as a valuable draft for team refinement.

RQ4 examines GenAI's role in task assignment and workload balancing. The GenSP tool's approach to assigning tasks results in more balanced workloads, as seen in the comparison with MyFlavoria's sprint history. Survey responses support this finding, with participants noting that GenAI could assist in distributing work fairly, but they also highlighted challenges in assessing individual skills and preferences purely through AI.

Finally, RQ5 considers the overall impact of GenAI on the effectiveness, efficiency, and quality of sprint planning. Both the GenSP evaluation and the survey

indicated that GenAI improves sprint planning, enhances backlog and task clarity, and supports better workload management. The majority of survey participants agreed that GenAI made the sprint planning process more effective, especially in saving time and reducing manual effort. However, the participants emphasize the continued need for human oversight.

In summary, by mapping each research question to the findings from GenSP’s tool-based evaluation and the practitioner survey, this study demonstrates that GenAI holds significant promise for addressing many of the persistent challenges in Scrum sprint planning. This also highlights the areas where further development in GenAI models is needed for performing more complex scenarios in sprint planning.

6.4 Discussion

The findings of this thesis offer valuable insights into the practical application and impact of GenAI in sprint planning. This section discusses these results in detail.

6.4.1 Evaluating GenSP Tool Data

When comparing GenSP-generated sprint plans with the MyFlavoria project’s historical sprint data, several advantages become evident. The GenAI-assisted plans are more closely aligned with sprint goals. The tool also demonstrates improved workload balancing and prompted clearer backlog definitions. GenSP’s structured approach to selecting user stories, estimating effort, and assigning tasks reduced the likelihood of overcommitment and made hidden dependencies more visible during planning. This tool lays a foundation for the feasibility of using GenAI directly in the sprint planning process.

6.4.2 Interpretation of Survey Results

The survey results indicate a generally positive attitude among experienced software developers toward using GenAI in sprint planning. The majority of participants recognize GenAI's value in automating backlog refinement, generating user stories, and supporting task breakdown. In particular, respondents highlighted time savings, reduced manual effort, and the ability to surface suggestions or dependencies that might otherwise be overlooked. In summary, the experienced scrum practitioners mostly support the usefulness of using GenAI in sprint planning.

However, the results also reveal a note of caution. While most participants would be comfortable using AI-generated sprint backlogs and task breakdowns as a starting point, there remains some skepticism about relying solely on AI for complex estimations or decisions that require deep contextual knowledge.

6.4.3 Key Contribution

This research makes certain key contributions to the study of GenAI in sprint planning. First, it offers a comprehensive review of the current challenges in traditional sprint planning. This thesis also highlights the gaps in existing research regarding the integration of GenAI into this process.

A key practical contribution is the design and development of GenSP. The tool provides ChatGPT-supported sprint backlog generation, story point estimation, task breakdown, and task distribution.

Additionally, the thesis contributes empirical insights by a structured survey of experienced Scrum practitioners. The survey results, along with the qualitative and quantitative analyses, provide evidence of the perceived value, limitations, and potential of GenAI in supporting sprint planning.

6.4.4 Limitations

There are several limitations in the thesis that should be acknowledged. First, the evaluation of the GenSP tool is not conducted with a real Scrum team in a live development environment. Instead, the tool is tested using the backlog and sprint data of an existing software project. Using a real project would show more closely how GenAI supports the sprint planning process.

Second, the number of survey respondents was limited. Although all participants have prior experience in software development and Agile methodologies, they did not use the GenSP tool directly. As a result, their responses reflect expert opinions based on their knowledge and understanding, rather than hands-on experience with the tool. While this helped ensure relevance, it may also have introduced some bias and limited the diversity of perspectives represented in the results. Larger-scale studies involving a broader range of teams and organizations would strengthen the generalizability of the findings.

Lastly, GenSP depends on the OpenAI GPT API, so its outputs are subject to the limitations of LLMs, such as hallucination [43], limited domain knowledge, or context misinterpretation, despite efforts to improve reliability through prompt engineering.

Despite these limitations, the thesis lays a foundation for further exploration and highlights important considerations for both researchers and practitioners interested in integrating GenAI into Agile project management.

7 Conclusion

This thesis explored how Generative AI (GenAI) can support and improve sprint planning in Agile software development. Using a literature review, prototype tool development, and a survey of Agile practitioners, the research found that GenAI can aid the sprint planning process.

The GenSP tool, created for this research, showed that GenAI can automate key tasks such as generating sprint backlogs based on team goals and capacity, estimating story points, breaking down user stories into actionable tasks, and assigning work more evenly. These features were tested using real project data and compared to traditional planning methods.

Survey responses from experienced Agile practitioners highlighted the practical value of GenAI. Most agreed that it can save time, provide helpful suggestions, and improve task clarity. However, they also noted challenges, such as understanding complex requirements, assessing individual skills, and ensuring security and privacy. Importantly, respondents emphasized that GenAI should support—not replace—human decision-making.

The results suggest that GenAI can make sprint planning more efficient and consistent, reducing manual effort and facilitating better team discussions. Nonetheless, human oversight remains essential for interpreting AI recommendations and maintaining alignment with team objectives.

7.1 Future Work

Although this thesis demonstrated the potential of GenAI in supporting Agile sprint planning, several avenues remain open for further exploration and improvement.

First, a key next step is to test GenAI-powered tools like GenSP in live, ongoing projects managed with Scrum. Evaluating the tool during active sprints would provide richer insights into its practical value, user acceptance, and impact on team performance in real-world settings. In addition, conducting surveys among a wider group of practitioners from diverse organizations and backgrounds would further strengthen the understanding of GenAI's applicability and effectiveness in various Agile environments.

Second, integrating GenSP into the Sprint Retrospective process could offer a feedback loop that continually improves AI-generated outputs. By allowing teams to review, discuss, and rate the accuracy and usefulness of GenAI suggestions during retrospectives, the system can adapt to each team's unique context and learn from previous sprints.

Finally, future research should compare different LLMs to identify the most effective one for Agile sprint planning. Assessing models based on accuracy, response quality, explainability, and integration ease will help determine which LLMs best meet the needs of Scrum teams.

Exploring these directions will strengthen the practical application of GenAI in Agile environments and contribute to the development of more adaptive and intelligent project management solutions that support Agile practices.

References

- [1] M. U. Haque, I. Dharmadasa, Z. T. Sworna, R. N. Rajapakse, and H. Ahmad, ““i think this is the most disruptive technology”: Exploring sentiments of chat-gpt early adopters using twitter data”, *arXiv preprint arXiv:2212.05856*, 2022.
- [2] V. Corso, L. Mariani, D. Micucci, and O. Riganelli, “Assessing ai-based code assistants in method generation tasks”, in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, 2024, pp. 380–381.
- [3] B. Sherifi, K. Slhoub, and F. Nembhard, “The potential of llms in automating software testing: From generation to reporting”, *arXiv preprint arXiv:2501.00217*, 2024.
- [4] C. Ebert and P. Louridas, “Generative ai for software practitioners”, *IEEE Software*, vol. 40, no. 4, pp. 30–38, 2023.
- [5] H. K. Flora and S. V. Chande, “A systematic study on agile software development methodologies and practices”, *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3626–3637, 2014.
- [6] M. Hron and N. Obwegeser, “Why and how is scrum being adapted in practice: A systematic review”, *Journal of Systems and Software*, vol. 183, p. 111 110, 2022.

-
- [7] Scrum.org, *What is scrum?*, <https://www.scrum.org/learning-series/what-is-scrum>, Accessed: 2024-06-27, 2020.
- [8] S. Alliance, “Sprint planning meeting: Why it’s critical for agile success”, *Scrum Alliance Resource Library*, 2025, Reviewed by: Bernie Maloney, Madhur Kathuria, and Raúl Herranz. [Online]. Available: <https://resources.scrumalliance.org/Article/sprint-planning-meeting>.
- [9] K. Schwaber and J. Sutherland, *The scrum guide*, Accessed: 2024-06-27, 2020. [Online]. Available: <https://scrumguides.org/scrums-guide.html>.
- [10] K. Melnyk, V. Hlushko, and N. Borysova, “Decision support technology for sprint planning”, *Radio Electronics, Computer Science, Control*, no. 1, pp. 135–145, 2020.
- [11] M. Golfarelli, S. Rizzi, and E. Turrinchia, “Multi-sprint planning and smooth replanning: An optimization model”, *Journal of systems and software*, vol. 86, no. 9, pp. 2357–2370, 2013.
- [12] M. Žáček, A. Hamplová, J. Tyrychtr, and I. Vrana, “Improvements for the planning process in the scrum method”, *Applied Sciences*, vol. 15, no. 1, p. 202, 2024.
- [13] L. Alsaber, E. Al Elsheikh, S. Aljumah, and N. S. M. Jamail, ““scumbear” framework for solving traditional scrum model problems”, *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 319–326, 2021.
- [14] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research”, *MIS quarterly*, pp. 75–105, 2004.
- [15] M. Kasunic, *Designing an effective survey*, 2005.
- [16] B. Wang et al., “Towards understanding chain-of-thought prompting: An empirical study of what matters”, *arXiv preprint arXiv:2212.10001*, 2022.

-
- [17] E. Guglielmi, V. Arnoudova, G. Bavota, R. Oliveto, and S. Scalabrino, “How do copilot suggestions impact developers’ frustration and productivity?”, *arXiv preprint arXiv:2504.06808*, 2025.
- [18] C. Stryker and E. Kavlakoglu, *What is artificial intelligence (ai)?*, Published 9 August 2024; accessed 2025-06-14, 2024. [Online]. Available: <https://www.ibm.com/think/topics/artificial-intelligence>.
- [19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd. Upper Saddle River, NJ: Prentice Hall, 2010, ISBN: 978-0-13-604259-4.
- [20] M. E. Klontzas, S. C. Fanni, and E. Neri, Eds., *Introduction to Artificial Intelligence (Imaging Informatics for Healthcare Professionals)*. Springer, 2023, ISBN: 978-3-031-25927-2. [Online]. Available: <https://doi.org/10.1007/978-3-031-25928-9>.
- [21] S. M. Mian, M. S. Khan, M. Shawez, and A. Kaur, “Artificial intelligence (ai), machine learning (ml) & deep learning (dl): A comprehensive overview on techniques, applications and research directions”, in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, IEEE, 2024, pp. 1404–1409.
- [22] A. Dubey, “Usage of deep learning in recent applications”, *Archives of Materials Science and Engineering*, vol. 115, no. 2, 2022.
- [23] S. S. Sengar, A. B. Hasan, S. Kumar, and F. Carroll, “Generative artificial intelligence: A systematic review and applications”, *Multimedia Tools and Applications*, pp. 1–40, 2024.
- [24] Z. Sordo, E. Chagnon, and D. Ushizima, “A review on generative ai for text-to-image and image-to-image generation and implications to scientific images”, *arXiv preprint arXiv:2502.21151*, 2025.

-
- [25] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, “Generative ai”, *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024.
- [26] L. Banh and G. Strobel, “Generative artificial intelligence”, *Electronic Markets*, vol. 33, no. 1, p. 63, 2023.
- [27] R. K. Gatla, A. Gatla, P. Sridhar, D. G. Kumar, and D. N. M. Rao, “Advancements in generative ai: Exploring fundamentals and evolution”, in *2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC)*, IEEE, 2024, pp. 1–5.
- [28] Stack Overflow. “Stack overflow developer survey 2024: AI developer tools”. Accessed: 2024-06-27, Stack Overflow, Accessed: Jun. 27, 2024. [Online]. Available: <https://survey.stackoverflow.co/2024/ai#developer-tools>.
- [29] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, “The impact of ai on developer productivity: Evidence from github copilot”, *arXiv preprint arXiv:2302.06590*, 2023.
- [30] Z. Zheng, K.-Y. Chen, X.-Y. Cao, X.-Z. Lu, and J.-R. Lin, “Llm-funcmapper: Function identification for interpreting complex clauses in building codes via llm”, *arXiv preprint arXiv:2308.08728*, 2023.
- [31] G. Bandarupalli, “Code reborn ai-driven legacy systems modernization from cobol to java”, *arXiv preprint arXiv:2504.11335*, 2025.
- [32] Y. Fu et al., “Creativity in the age of ai: Evaluating the impact of generative ai on design outputs and designers’ creative thinking”, *arXiv preprint arXiv:2411.00168*, 2024.
- [33] S. Dandotiya, “Generative ai for software testing: Harnessing large language models for automated and intelligent quality assurance [j]”, *International Journal of Science and Research Archive*, vol. 14, no. 1, pp. 1931–1935, 2025.

-
- [34] C. Arora, J. Grundy, and M. Abdelrazek, “Advancing requirements engineering through generative ai: Assessing the role of llms”, in *Generative AI for Effective Software Development*, Springer, 2024, pp. 129–148.
- [35] T. Rahman and Y. Zhu, “Automated user story generation with test case specification using large language model”, *arXiv preprint arXiv:2404.01558*, 2024.
- [36] T. Dyba and T. Dingsoyr, “What do we know about agile software development?”, *IEEE software*, vol. 26, no. 5, pp. 6–9, 2009.
- [37] A. Alliance, *Principles behind the agile manifesto*, <https://agilemanifesto.org/principles.html>, 2001.
- [38] M. Coram and S. Bohner, “The impact of agile methods on software project management”, in *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS’05)*, IEEE, 2005, pp. 363–370.
- [39] N. Damij and T. Damij, “An approach to optimizing kanban board workflow and shortening the project management plan”, *IEEE transactions on engineering management*, vol. 71, pp. 13 266–13 273, 2021.
- [40] A. Putta, M. Paasivaara, and C. Lassenius, “Benefits and challenges of adopting the scaled agile framework (safe): Preliminary results from a multivocal”, in *International Conference on Product-Focused Software Process Improvement, Vol. 1*, Springer International Publishing, 2018.
- [41] M. Hron and N. Obwegeser, “Scrum in practice: An overview of scrum adaptations”, in *Hawaii International Conference on System Sciences*, Curran Associates, Inc., 2018, pp. 4496–4505.

- [42] M. A. Beltran, M. I. Ruiz Mondragon, and S. H. Han, “Comparative analysis of generative ai risks in the public sector”, in *Proceedings of the 25th Annual International Conference on Digital Government Research*, 2024, pp. 610–617.
- [43] M. T. Hicks, J. Humphries, and J. Slater, “Chatgpt is bullshit”, *Ethics and Information Technology*, vol. 26, no. 2, pp. 1–10, 2024.
- [44] B. Cabrero-Daniel, “How reliance on genai might limit human creativity and critical thinking in requirements engineering”, 2025.
- [45] M. Abbas, F. A. Jam, and T. I. Khan, “Is it harmful or helpful? examining the causes and consequences of generative ai usage among university students”, *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, p. 10, 2024.
- [46] Scrum.org. “What is sprint planning?”, Scrum.org. [Online]. Available: <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-sprint-planning>.
- [47] Atlassian. “How to plan a sprint”. Accessed: 2024-06-27. [Online]. Available: <https://www.atlassian.com/agile/scrum/sprint-planning>.
- [48] R. K. Mallidi and M. Sharma, “Study on agile story point estimation techniques and challenges”, *Int. J. Comput. Appl*, vol. 174, no. 13, pp. 9–14, 2021.
- [49] D. Özkan and A. Mishra, “Agile project management tools: A brief comparative view”, *Cybernetics and Information Technologies*, vol. 19, no. 4, pp. 17–25, 2019.
- [50] M. O. Zander and M. Meboldt, “Navigating the unknown: Introduction of an analysis metric for the sprint plan and sprint review in educational agile development projects”, in *Proceedings of the 52nd Annual Conference of SEFI, Lausanne, Switzerland.*, Zenodo, 2024.

-
- [51] L. A. Garcia, E. Oliveira Jr, and M. Morandini, “Tailoring the scrum framework for software development: Literature mapping and feature-based support”, *Information and Software Technology*, vol. 146, p. 106 814, 2022.
- [52] S. Yermolaieva, “Communication challenges in agile teams from the communication theory prospective”, in *Proceedings of the 2020 European Symposium on Software Engineering*, 2020, pp. 88–95.
- [53] H. K. Dam, T. Tran, J. Grundy, A. Ghose, and Y. Kamei, “Towards effective ai-powered agile project management”, in *2019 IEEE/ACM 41st international conference on software engineering: new ideas and emerging results (ICSE-NIER)*, IEEE, 2019, pp. 41–44.
- [54] A. Bahi, J. GHARI, and Y. Gahi, “Integrating generative ai for advancing agile software development and mitigating project management challenges.”, *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 3, 2024.
- [55] S. Ostrowski, “Using genai in it project management: Case studies, insights and challenges.”, *Scientific Papers of Silesian University of Technology. Organization & Management/Zeszyty Naukowe Politechniki Slaskiej. Seria Organizacji i Zarzadzanie*, no. 215, 2025.
- [56] M. Choetkiertikul, H. K. Dam, T. Tran, T. Pham, A. Ghose, and T. Menzies, “A deep learning model for estimating story points”, *IEEE Transactions on Software Engineering*, vol. 45, no. 7, pp. 637–656, 2018.
- [57] R. K. Mallidi, M. Sharma, and Y. P. Paladugu, “Story point estimate model: Project development using generative ai (genai) tools”, *International Journal of Computer Applications*, vol. 975, p. 8887, 2024.
- [58] M. R. Islam and P. Sandborn, “Multimodal generative ai for story point estimation in software development”, *arXiv preprint arXiv:2505.16290*, 2025.

-
- [59] T. K. Chiu, “The impact of generative ai (genai) on practices, policies and research direction in education: A case of chatgpt and midjourney”, *Interactive Learning Environments*, vol. 32, no. 10, pp. 6187–6203, 2024.
- [60] *Myflavoria application*, <https://myflavoria.fi/app/>, Accessed: 22 July 2025, 2025.
- [61] F. K. Willits, G. L. Theodori, and A. Luloff, “Another look at likert scales”, *Journal of rural social sciences*, vol. 31, no. 3, p. 6, 2016.
- [62] Amazon Web Services, *What is a RESTful API?*, <https://aws.amazon.com/what-is/restful-api/>, Accessed on July 04, 2025.
- [63] Express.js, *Express.js 5.x API Reference*, <https://expressjs.com/en/5x/api.html>, Accessed on July 04, 2025.
- [64] MongoDB Inc., *MongoDB Documentation*, <https://www.mongodb.com/docs/>, Accessed on July 04, 2025.
- [65] MongooseJS, *Mongoose Documentation*, <https://mongoosejs.com/docs/index.html>, Accessed on July 04, 2025.
- [66] OpenAI, *OpenAI Node.js Library Documentation*, <https://platform.openai.com/docs/libraries/node-js-library>, Accessed on July 04, 2025.
- [67] M. Principe and D. Yoon, “A web application using mvc framework”, in *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)*, The Steering Committee of The World Congress in Computer Science, Computer . . . , 2015, p. 10.
- [68] OpenAI. “Api reference: Chat”, OpenAI, Accessed: Jul. 25, 2025. [Online]. Available: <https://platform.openai.com/docs/api-reference/chat>.
- [69] P. Bansal, “Prompt engineering importance and applicability with generative ai”, *Journal of Computer and Communications*, vol. 12, no. 10, pp. 14–23, 2024.

Appendix A Survey Questions

The survey questions are available here: <https://forms.gle/sE6fubxgCpDKCVtY9>

A.1 Generative AI in Sprint Planning: Survey Questions

1. Contact Email

2. Country of Residence

3. Your Current Role

- Software Engineer/Developer
- DevOps Engineer
- Quality Assurance Engineer (QA)
- Scrum Master
- Product Owner
- Other

4. How many years of professional software development experience do you have?

- 0–2 years

- 3–5 years
- 6–10 years
- More than 10 years

5. What is your familiarity with Agile/Scrum methodologies?

- Not familiar (no direct experience)
- Beginner (some exposure)
- Intermediate (regular team member)
- Advanced (Scrum Master/Product Owner)

6. Do you or have you ever used project management tools? (e.g., Jira, Trello, etc.)

- Yes
- No

7. How frequently do you use generative AI tools (e.g., ChatGPT, Gemini, Copilot) in your work?

- Frequently (on most days or in most projects)
- Occasionally (a few times a month)
- Rarely (a few times total)
- Never

8. Have you ever used generative AI as part of your Sprint Planning process?

- Yes, regularly
- Yes, occasionally

- No, never

9. **Which sprint planning activities have you used generative AI for?**

- Writing or refining user story descriptions
- Estimating story points or effort for user stories
- Breaking down user stories into tasks/sub-tasks
- Prioritizing or grooming the product backlog
- Drafting or clarifying sprint goals
- I have not used GenAI in any sprint planning activities
- Other

10. **How valuable do you think generative AI could be in helping to create the sprint backlog (suggesting user stories based on sprint goal, capacity, etc.)? (1 = Not Valuable, 5 = Extremely Valuable)**

11. **How valuable could generative AI be in identifying missing details, dependencies, or acceptance criteria for sprint backlog items? (1 = Not Valuable, 5 = Extremely Valuable)**

12. **How helpful could generative AI be in suggesting prioritization or ordering of items within the sprint backlog? (1 = Not Helpful, 5 = Very Helpful)**

13. **In your opinion, during sprint backlog creation, which area could benefit the most from generative AI?**

- Sprint backlog item suggestions
- Backlog item refinement
- Reprioritizing backlog items

- Other

14. **What do you see as the biggest challenge in using generative AI for sprint backlog creation?**

- Ensuring sprint backlog alignment with sprint goal
- Quality/Relevance of AI suggestions
- Understanding complex business logic/dependencies
- Other

15. **Your Thoughts on Applying GenAI to Sprint Backlog Creation** (Open comment)

16. **"Generative AI can estimate user story points based on story title, descriptions, and historical data." How much do you agree?** (1 = Strongly Disagree, 5 = Strongly Agree)

17. **"AI-proposed estimates or insights help our team reach consensus on story point estimations more quickly." Rate your agreement.** (1 = Strongly Disagree, 5 = Strongly Agree)

18. **Would you be comfortable using an AI-generated estimate as a starting point for team discussion?** (1 = Not Comfortable, 5 = Very Comfortable)

19. **What do you see as the biggest challenge in using generative AI for story point estimation?**

- Quality/Accuracy of estimates
- Handling novel or complex stories
- Other

20. **Your Thoughts on Using GenAI for Estimating Story Points in Sprint Planning** (Open comment)
21. **“Generative AI is helpful in breaking down user stories into a clear list of tasks or sub-tasks.” How much do you agree?** (1 = Strongly Disagree, 5 = Strongly Agree)
22. **How helpful could generative AI be in identifying potential task dependencies (within the story or to other stories)?** (1 = Not Helpful, 5 = Very Helpful)
23. **Would you be comfortable using a generative AI-generated task breakdown as a starting point for the team to refine?** (1 = Not Comfortable, 5 = Very Comfortable)
24. **What do you see as the biggest challenge in using generative AI for story-to-task breakdown?**
- Generating overly generic or irrelevant tasks
 - Understanding the technical implementation approach
 - Lack of technical detail in the user story
 - Other
25. **Your Thoughts on Using GenAI for Breaking Down User Stories into Tasks** (Open comment)
26. **How helpful could generative AI be in suggesting task assignments based on individual skills, experience, or development goals?** (1 = Not Helpful, 5 = Very Helpful)
27. **How valuable could generative AI be in helping to balance the workload across team members during sprint planning?** (1 = Not Valuable,

5 = Extremely Valuable)

28. **What do you see as the biggest challenge in using generative AI for task distribution?**

- Accurately assessing individual skills/capabilities
- Handling team preferences in task distribution
- Perceived fairness/bias in suggestions
- Other

29. **Your Thoughts on Using GenAI for Task Distribution in Sprint Planning** (Open comment)

30. **“Generative AI, on the whole, can improve the effectiveness of our sprint planning process.” Please indicate your agreement.** (1 = Strongly Disagree, 5 = Strongly Agree)

31. **In your opinion, which aspect of sprint planning benefits the most from generative AI support?**

- User story refinement
- Sprint backlog creation
- Story point estimation
- Task breakdown and planning
- Backlog prioritization
- Improving planning efficiency
- Other

32. **What challenges or concerns do you have (or do you foresee) with using generative AI in sprint planning?**

- AI suggestions can be inaccurate
- AI might reduce team discussion/human insight
- Not convinced AI adds value
- Difficulty integrating with existing tools
- Privacy and security concerns
- Other

33. **Your Thoughts on Using GenAI in Sprint Planning Overall** (Open comment)