

Context-Aware Automated Sprint Plan Generation for Agile Software Development

Kula, Elvan; Van Deursen, Arie; Gousios, Georgios

DOI

[10.1145/3691620.3695540](https://doi.org/10.1145/3691620.3695540)

Publication date

2024

Document Version

Final published version

Published in

Proceedings - 2024 39th ACM/IEEE International Conference on Automated Software Engineering, ASE 2024

Citation (APA)

Kula, E., Van Deursen, A., & Gousios, G. (2024). Context-Aware Automated Sprint Plan Generation for Agile Software Development. In *Proceedings - 2024 39th ACM/IEEE International Conference on Automated Software Engineering, ASE 2024* (pp. 1745-1756). (Proceedings - 2024 39th ACM/IEEE International Conference on Automated Software Engineering, ASE 2024). ACM.
<https://doi.org/10.1145/3691620.3695540>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Context-Aware Automated Sprint Plan Generation for Agile Software Development

Elvan Kula
Delft University of Technology
Delft, The Netherlands
e.kula@tudelft.nl

Arie van Deursen
Delft University of Technology
Delft, The Netherlands
arie.vandeursen@tudelft.nl

Georgios Gousios
Delft University of Technology
Delft, The Netherlands
g.gousios@tudelft.nl

ABSTRACT

Sprint planning is essential for the successful execution of agile software projects. While various prioritization criteria influence the selection of user stories for sprint planning, their relative importance remains largely unexplored, especially across different project contexts. In this paper, we investigate how prioritization criteria vary across project settings and propose a model for generating sprint plans that are tailored to the context of individual teams. Through a survey conducted at ING, we identify urgency, sprint goal alignment, and business value as the top prioritization criteria, influenced by project factors such as resource availability and client type. These results highlight the need for contextual support in sprint planning. To address this need, we develop an optimization model that generates sprint plans aligned with the specific goals and performance of a team. By integrating teams' planning objectives and sprint history, the model adapts to unique team contexts, estimating prioritization criteria and identifying patterns in planning behavior. We apply our approach to real-world data from 4,841 sprints at ING, demonstrating significant improvements in team alignment and sprint plan effectiveness. Our model boosts team performance by generating plans that deliver more business value, align more closely with sprint goals, and better mitigate delay risks. Overall, our results show that the efficiency and outcomes of sprint planning practices can be significantly improved through the use of context-aware optimization methods.

CCS CONCEPTS

• **Software and its engineering** → **Software development process management**;

KEYWORDS

agile methods, sprint planning, context-aware optimization

ACM Reference Format:

Elvan Kula, Arie van Deursen, and Georgios Gousios. 2024. Context-Aware Automated Sprint Plan Generation for Agile Software Development. In *39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24)*, October 27–November 1, 2024, Sacramento, CA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3691620.3695540>



This work is licensed under a Creative Commons Attribution International 4.0 License. ASE '24, October 27–November 1, 2024, Sacramento, CA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1248-7/24/10.
<https://doi.org/10.1145/3691620.3695540>

1 INTRODUCTION

Effective planning is crucial for the successful execution of software development projects [14]. Central to the planning is the ability to prioritize and select software features that deliver the most value to customers while mitigating delays. Over the past two decades, agile methodologies have become increasingly popular for managing software projects [15, 34]. Agile uses an iterative approach, enabling teams to manage changing priorities, rapidly deliver business value, and inherently reduce risks. However, effective planning remains challenging in agile settings. Previous research indicates that nearly half of agile projects exceed their timelines by 25% [66] and deliver 56% less business value than anticipated [6], highlighting the need for improved planning strategies.

In agile settings, software is developed incrementally through short iterations known as *sprints* [10]. Each sprint involves completing a subset of requirements, expressed as *user stories* [16]. Before a sprint begins, the team performs *sprint planning* to define the sprint goal and select user stories from the backlog. Various factors, referred to as *prioritization criteria*, such as business value and urgency, are used to prioritize and select user stories for sprint planning [18, 30, 49, 64]. Although business value is typically considered the main prioritization criterion in agile methods, previous research [51, 54] suggests that this may not always reflect actual practice. Sprint plans are developed by teams based on their cumulative knowledge and biases, making them specific to the context of each team. The relative impact of the prioritization criteria remains largely unexplored, especially across different project contexts.

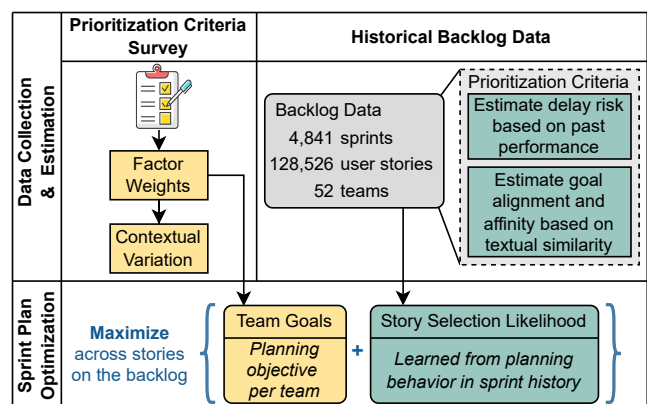


Figure 1: Overall study design: We collect survey data (shown in yellow) to obtain teams' weightings of prioritization criteria and their planning objectives. We collect historical backlog data (shown in gray) and use machine learning techniques (visualized in green) to estimate prioritization criteria. We develop an optimization model, integrating teams' planning objectives and behavior, to generate sprint plans tailored to each team's specific context.

The sprint planning process is complex and time-consuming, particularly for large projects where backlogs can grow to hundreds of user stories [52]. Agile teams would benefit from automated support that estimates prioritization criteria and generates sprint plans tailored to their specific context. Existing models [7, 29, 37] generate sprint plans based on team estimates of prioritization criteria, aiming to maximize business value. Some studies [3, 50] have extended these models to consider additional objectives, such as maximizing sprint goal alignment and capacity usage. However, existing approaches rely on team estimates of prioritization criteria and do not account for contextual influences. Recent studies (e.g., [13, 44]) suggest that machine learning techniques can improve software project management by providing contextual support and insights from project data. This has the potential to enhance the efficiency and outcomes of sprint planning.

The goal of this paper is to develop a model for generating sprint plans that align with the specific goals and performance of individual teams. To achieve this, we conducted a case study at ING, a large Dutch internationally operating bank, following the study design outlined in Figure 1. We start by investigating how prioritization criteria affect the selection of user stories for sprint planning. We conduct a survey with 52 teams to assess how they weigh the importance of these criteria and how this is influenced by project characteristics. Next, we collect historical backlog data from 4,841 sprints and use machine learning techniques to estimate prioritization criteria. We then develop an optimization model that integrates teams' planning objectives and sprint history to generate sprint plans tailored to each team's specific context. The model learns from past team performance to identify and incorporate planning behavior patterns. We evaluate our model through both quantitative and qualitative analyses. For the quantitative analysis, we use the historical backlog data to assess the model's effectiveness and alignment with team planning. We compare the performance of our model to the state-of-the-art in automated sprint planning methods. For the qualitative analysis, we interview teams to gather insights into their perceptions of the model's usability.

Our survey results show that urgency, sprint goal alignment, and business value are the most important prioritization criteria, with their influence depending on project characteristics, such as project resources, priority, client type, and security level. The quantitative evaluation of our model demonstrates significant improvements in team alignment and sprint plan effectiveness. On average, the model achieves an 88% overlap in selected stories with the team's actual sprint plans, and a 74% semantic relatedness between differing stories. Our model outperforms the state-of-the-art and improves team performance by generating sprint plans that deliver 29% more business value, exhibit 14% stronger alignment with sprint goals, and reduce delay risk by 42%. In the qualitative evaluation, the majority of teams found our approach to be consistent with their goals and valuable as interactive support.

The main contributions of this paper are:

- A set of prioritization criteria ordered by their importance for sprint planning (Section 3.3).
- A context-aware optimization approach to generate sprint plans that align with team goals and performance (Section 4).
- An empirical evaluation of the approach and comparison to the state-of-the-art, demonstrating significant improvements in team alignment and sprint plan effectiveness (Section 5.3–5.4).
- A qualitative analysis of the approach with software teams identifying areas for future research (Section 5.5).

2 BACKGROUND AND RELATED WORK

2.1 Sprint Planning

In agile software projects, *sprints* typically span 2–4 weeks [10]. During this period, teams design, implement, test, and deliver a product increment, such as a working milestone. Each sprint requires the completion of a set of *user stories*, which are brief descriptions of features written from the perspective of the end user. Teams maintain a prioritized list of pending user stories, known as the *product backlog* [59]. The product owner organizes the backlog by sorting user stories according to their urgency, ensuring that the most urgent stories are prioritized at the top. The urgency of a story is determined based on immediate customer needs and project deadlines. Each user story includes a title, a textual description clarifying the task, and several standard fields detailing the story's type, business value, and dependencies.

Before each sprint, teams hold a *sprint planning* meeting, divided into two parts. In the first part, the product owner and development team establish the sprint goal and discuss user stories in the backlog. The sprint goal is a concise statement that describes the primary focus of the sprint, such as implementing a new feature. In the second part of the meeting, the team breaks down user stories into specific tasks and estimates the effort required for each user story. Teams rely on expert judgement [65] to estimate effort, often using *story points* as the unit of measure. Story points reflect the relative effort, complexity, and risks associated with a user story [17]. The team then selects user stories to implement in the upcoming sprint based on prioritization criteria (described in Section 3.1). These criteria consider the potential value of the user stories and the risks associated with their execution. Additionally, the selection process accounts for development constraints, such as dependencies and the team's delivery capacity. Teams measure their capacity for future sprints by monitoring their *velocity*, which represents the average number of story points completed in previous sprints [17].

2.2 Agile development at ING

We conduct our research in the context of ING, a large Dutch internationally operating bank with more than 15,000 developers. In recent years, ING has reinvented its organisational structure, moving from traditional functional departments to a fully agile setup inspired by Spotify's '*Squads, Tribes and Chapters*' model [41]. We conducted our case study at ING TECH, the IT department responsible for the bank's main applications used by millions of customers. This department has significant variety in terms of products and application domains. The teams develop banking applications for customers, as well as cloud software and software tools for internal use. All teams follow Scrum [10] as agile methodology and work in sprints of one to four weeks. Each team consists of 5 to 9 members, including a product owner. To estimate the effort required for user stories, teams use planning poker [32] to assign story points.

2.3 Related Work

Research on automated sprint planning (e.g., [7, 29, 37]) treats the process as an optimization problem. Previous studies have used methods such as mathematical programming [7, 29, 37] and genetic algorithms [3, 50] to select the optimal set of user stories for a sprint. These approaches aim to maximize business value over the selection of user stories, relying on team estimates of prioritization criteria and using a weighted sum of these criteria as the objective function. Golfarelli et al. [29] developed an extensive objective function that incorporates criticality risk and affinity as factors influencing the business value of user stories. Their work represents the state-of-the-art benchmark, which we use for our model evaluation in Section 5. Al-Zubaidi et al. [3] and Ozcelikkan et al. [50] extended their focus to multi-objective optimization, incorporating additional objectives such as maximizing sprint goal alignment and capacity usage. While most efforts address planning for individual sprints, recent studies (e.g., [30, 50]) have developed optimization models for multi-sprint plans that allow for re-planning during execution.

Despite these advancements, existing approaches have two main limitations: 1. They use generic objective functions that are not tailored to individual team contexts. 2. They are not fully automated, as they rely on team estimates of prioritization criteria. Our study addresses these gaps by providing automated, contextual support for generating sprint plans. Our model incorporates teams' planning objectives and sprint history to create plans aligned with team goals and performance. We use machine learning techniques to estimate prioritization criteria, thereby relieving teams from the task of manual estimation. To evaluate the performance of our model, we use extensive real-world data from our case company.

3 PRIORITIZATION CRITERIA SURVEY

We start by identifying prioritization criteria (i.e., factors that influence the prioritization and selection of user stories for sprint planning) through literature analysis and observations at the case company. We aim to address the following research questions:

- **RQ1.1 Factor weights:** How do teams weigh the importance of prioritization criteria for sprint planning?
- **RQ1.2 Weight variations:** How do project characteristics affect the weightings of prioritization criteria?

To answer these questions, we conduct a survey with 52 teams (Section 3.2), through which we assess the factor weights and how they vary across different project settings (Section 3.3).

3.1 Deriving Factors from Literature and ING

From our literature analysis and observations at the case company, we identified six prioritization criteria. These criteria specifically affect the *order* in which user stories are prioritized and selected for planning, excluding development constraints such as dependencies and team capacity. We derived five of these criteria from the literature, while “strategic alignment” emerged from discussions with teams at ING. Below, we explain these criteria in detail, with each factor name underlined:

- Business value refers to the potential impact of a user story on achieving business goals and satisfying customer needs [16, 53]. This factor is often considered the primary criterion in agile

methods [33, 42]. However, this assumption has been debated in previous research [51, 54], suggesting that it may not always reflect actual practice and requires further research.

- Urgency measures the time sensitivity of a user story, determined by its relevance to critical customer needs or alignment with project deadlines [1, 2].
- Sprint goal alignment evaluates how well user stories contribute to achieving the overall objective of the sprint [3, 25].
- Affinity measures the degree of similarity or relatedness between user stories within a project [7, 30]. High-affinity user stories share common themes or objectives, while low-affinity stories have less overlap in functionality or purpose. Delivering affine stories together in the same sprint can enhance the utility of the software functionality [29]. For example, a “data extraction” story may have limited value on its own but becomes more valuable when delivered with a “data loading” story. Although affine stories complement each other, they are not interdependent and can be implemented separately.
- Delay risk indicates the likelihood of a user story experiencing delays, influenced by factors such as complexity, uncertainty, and impact on existing functionality [44, 49, 64]. User stories with high delay risk threaten deadlines, potentially leading to incomplete work and postponed feature delivery to customers.
- Based on discussions with teams at ING, we introduce strategic alignment as a new factor. This criterion evaluates how well user stories align with the organization's long-term strategic goals. During sprint planning meetings at ING, we observed that teams prioritized stories aligning with the company's strategic goals to improve product viability and promote software reuse.

3.2 Survey Setup

3.2.1 Survey design. We developed an online survey to be completed collaboratively by software teams. The survey consisted of a mix of closed and open-ended questions; the final survey instrument is provided in the supplemental material [43]. To provide context, the survey's start page contained a brief outline of our study's purpose. The survey was divided into two main sections: the first section included multiple-choice questions to gather demographic data on the size, years of existence, geographic distribution, and application domains of the participating teams [38]. Additionally, teams were asked to provide their identification number from *ServiceNow*¹, the backlog management tool used at ING. This allowed us to link survey responses with project data to determine project settings.

In the second section, teams were asked to rank the prioritization criteria based on their importance for sprint planning in their current project. Teams were encouraged to discuss and determine the importance of the criteria collaboratively during a group meeting or at the start of a sprint planning session. They used a drag-and-drop format to rank the criteria, which were presented in random order to reduce ordering bias [62]. After ranking, teams were asked to collectively weigh the importance of each factor using a slider scale, with values ranging from 0 (not important) to 1 (highest

¹<https://www.servicenow.com/>

importance). We designed the slider scale to be intuitive and easy for teams to adjust their weights. To ensure relative weighting, the total score across all prioritization criteria was limited to 1. At the end of this section, we included an open-ended question allowing respondents to suggest additional factors. We received nine responses, which we reviewed manually and found to be either rephrasings or sub-cases of existing prioritization criteria.

3.2.2 Survey validation. We piloted the survey with five randomly selected teams from ING TECH to refine the questions [39]. The pilot version featured an additional open-ended question for feedback on the survey content. All five teams provided feedback, highlighting the need for slider scales to assign factor weights and revealing ambiguity in the factor names. Based on their input, we provided definitions for the prioritization criteria in the final survey.

3.2.3 Survey execution. Our target population consisted of all 301 software teams within ING TECH. We accessed a mailing list containing 115 product owners representing these teams. For the final survey, we excluded the five teams and their respective product owners involved in the pilot run. In June 2023, we distributed the survey to the remaining 112 product owners and their 296 teams. We sent personal invitation emails to the product owners, explaining the survey’s purpose and requesting them to complete the survey with their teams. Participants had a total of three weeks to respond. We received responses from 52 teams, resulting in a response rate of 17%. We sent reminders halfway through the second week to follow up on non-responders.

3.2.4 Survey demographics. The survey gathered demographic information about the teams. The majority of teams (92%) reported to consist of five to eight members. Team existence ranged from one year (18%) to over five years (21%), with a median of three years. Additionally, 86% of teams indicated having had the same product owner over the past year, and 32% reported being globally distributed, conducting sprint planning meetings online. The teams worked in diverse application domains: web (27%), mobile (18%), desktop (11%), cloud-based (21%), and AI/data science (23%).

3.2.5 Survey data analysis. To investigate how teams perceive the importance of prioritization criteria (RQ1.1), we visualize the distributions of rank order responses and use descriptive statistics to compare factor weights. To assess whether factor weights are affected by project characteristics (RQ1.2), we investigate the impact of two main attributes: project size and project type. Previous research suggests that these attributes can affect requirements engineering approaches in agile projects [9, 54]. For project size, we measure resource availability in terms of the number of people assigned to the project, the time duration, and the budget allocated. For project type, we assess the priority assigned to the project, whether the client is internal or external to the case company, and whether the project requires resource-intensive security testing. These characteristics are used at ING to classify projects by size and type for planning and resource allocation. An overview of the extracted project characteristics and their descriptions is provided in Table 2. Using team ID numbers from the survey responses, we link the respondents’ factor weightings to their corresponding projects in the backlog management data. We extract the project characteristics directly from the primitive attributes of the project in the data.

We then conduct a correlation analysis to determine how these project characteristics affect the assigned factor weights. Since our data is not normally distributed, we use Spearman’s rank correlation [61] and apply Holm’s correction [35] to adjust for multiple comparisons.

3.3 Survey Results

3.3.1 (RQ1.1) Factor Weights. Table 1 presents the distributions of rank order responses and factor weights assigned by respondents. The “Rank” column indicates the order of factors by their weighted average rank scores, ranging from rank 1 (most important) to rank 6 (least important). Urgency ranked first, sprint goal alignment second, and business value third, with weighted average rank scores of 2.72 or lower. Over 67% of teams ranked urgency and sprint goal alignment as the most or second most important factors, with median weight scores of 0.26 or higher. Less than half (48%) of the teams ranked business value as one of the top two factors, yet it received a high median weight score of 0.21. Affinity and delay risk were perceived as being less important, with weighted average rank scores of 4.33 or higher, and notably lower median weight scores of 0.12 or less. Strategic alignment was ranked as the least or second least important factor by 82% of teams. Further analysis shows consistent rankings for sprint goal alignment and strategic alignment, with standard deviations lower than 1.05. There was greater variability in the rankings for other factors, with standard deviations ranging from 1.15 to 1.23.

Urgency, sprint goal alignment, and business value are the top most important prioritization criteria. Each is perceived to contribute more than 20% to determining the priority of a story.

Table 1: Overview of the prioritization criteria and their perceived importance for sprint planning. Teams ranked the factors by importance and assigned weights using values from 0 (not important) to 1 (highest importance). Rank distribution shows the distributions of rank order responses, with rank 1 being the most important rank and rank 6 the least important. WA represents the weighted average of factor ranks. Median weight and 95% CI indicate the median and 95% confidence interval of the weights assigned to the factors. The overall Rank is determined by the order of the weighted averages.







Factor	Rank distribution						Median			Rank
	1	2	3	4	5	6	WA	weight	95% CI	
Urgency							1.92	0.33	[0.18, 0.45]	#1
Sprint goal alignment							2.09	0.26	[0.14, 0.41]	#2
Business value							2.72	0.21	[0.11, 0.38]	#3
Affinity							4.33	0.12	[0.04, 0.19]	#4
Delay risk							4.51	0.10	[0.04, 0.16]	#5
Strategic alignment							5.25	0.07	[0.02, 0.11]	#6

Table 2: Results of the correlation analysis between project characteristics and the weights assigned to the prioritization criteria in survey responses. The prioritization criteria are abbreviated as follows: urgency (UR), sprint goal alignment (SG), business value (BV), affinity (AF), delay risk (DR), and strategic alignment (SR). Spearman’s correlation [61] coefficients are used to show relationship strengths, depicted by colors indicating **weak, **moderate** or **strong** relationships. Statistical significance is indicated with * at the 0.05 level after Holm correction [35].**

Project attribute	Project characteristic	Description of how we measured the characteristic	Type	Spearman’s correlation coefficients					
				UR	SG	BV	AF	DR	SA
Project size	nr-people	Total number of people working on the project	Continuous	0.41*	0.53*	0.46*	-0.11	-0.18	0.29*
	time	Planned project duration in days	Continuous	-0.30*	0.27*	0.38*	-0.14	-0.49*	0.41*
	budget	Total estimated monetary project costs	Continuous	0.25	0.52*	0.58*	0.16	-0.32*	0.36*
Project type	priority	Assigned priority class: 1. low prio, 2. moderate prio, 3. high prio	Categorical	0.58*	0.43*	0.49*	0.18*	0.54*	-0.25*
	client-type	Whether the project’s client is external to ING	Binary	0.45*	0.37	0.67*	0.42*	0.51*	-0.39*
	security-level	Whether the project requires resource-intensive security testing	Binary	-0.54*	-0.44*	-0.42*	0.61*	0.45*	-0.38*

3.3.2 (RQ1.2) *Weight Variations*. Table 2 presents the results of the correlation analysis between project characteristics and the weights assigned to the prioritization criteria by survey respondents. Significant correlations indicate variations in factor importance across different project settings. Teams working on projects with abundant resources assign significantly higher weights to sprint goal alignment, business value, and strategic alignment. They show less concern for affinity and delay risk, likely because they have sufficient resources to mitigate the consequences of fragmented and late deliveries. Teams working on high-priority projects, with external clients, or on security-critical systems prioritize delay risk and affinity more, while assigning less weight to strategic alignment. Specifically, high-priority projects and external client projects prioritize the customer-focused criteria, urgency and business value, whereas security-critical projects place more emphasis on affinity, possibly for end-to-end testing.

The importance of prioritization criteria varies significantly based on project characteristics, such as resources, priority, client type, and security level. *This variation demonstrates the need for contextual support in sprint planning practices.*

4 MODELING STORY PRIORITIZATION AND SPRINT PLAN OPTIMIZATION

The variations in factor weights across teams highlight the need for contextual support in sprint planning. To address this need, we aim to develop a model that generates sprint plans tailored to the specific context of each team. We use survey responses and sprint history data to model each team’s planning objectives and past performance. First, we collect historical backlog data (Section 4.1) and apply machine learning techniques to estimate the prioritization criteria for user stories (Section 4.2). To capture planning behavior, we develop a machine learning model that learns from a team’s sprint history to predict the likelihood of the team selecting a particular story for their upcoming sprint (Section 4.3). We derive team planning objectives from the prioritization criteria and their corresponding weights provided in the survey responses (Section 4.4). Combining these elements, we build an optimization model based on linear programming (Section 4.5). As illustrated in Figure 1, the optimization is guided by an objective function composed of two components: the team’s planning objective, which reflects their goals, and a selection likelihood estimate, which reflects their

planning behavior. By optimizing this objective while adhering to development constraints, our model selects the optimal set of user stories tailored to the team’s context for the upcoming sprint.

4.1 Backlog Data Collection

4.1.1 *Backlog data*. To develop and evaluate our model, we require a dataset containing historical records of each team’s backlog and sprints. For each sprint, this dataset should include the *identification number*, *start date*, *end date*, *team velocity*, the textual *sprint goal* field, and the set of user stories selected for that sprint. Similarly, user stories should include their *identification number*, *urgency*, *business value*, *story points*, *dependencies*, and the textual *title* and *description* fields. Since story contents might change before a sprint begins, we capture the information recorded on the day of sprint planning. This ensures consistency with the data available to the team during planning. For each team, we extract snapshots of their backlog on the days of sprint planning, linking team ID numbers with user stories to obtain the list of stories available. If a story is associated with a sprint ID number, it indicates that the story has been planned; if not, it remains unplanned in the backlog.

At ING, we extracted log data from the backlog management tool *ServiceNow*. This dataset contains records from 4,841 sprints and 128,526 user stories from the 52 respondent teams, covering the period from January 1, 2019 to January 1, 2023.

4.1.2 *Data pre-processing*. We took several steps to eliminate noise and address missing values in the data. First, we filtered out sprints with a status other than ‘Completed’, focusing on fully executed sprints. We then removed sprints that underwent significant content alterations during development, as these instances are likely unstable. After cleaning the data, the final dataset reduced to 4,812 sprints from 52 teams.

4.2 Estimating Prioritization Criteria

To model story prioritization, we need to measure or estimate the prioritization criteria for past user stories. We do this as follows:

- **Business value.** We extract business value directly from the primitive attributes of the stories in the dataset. This value is represented by a numerical score between 1 and 10, assigned by the product owner, indicating its perceived value to the customer. A higher score denotes greater business value.
- **Urgency:** Urgency is derived from the position of the story in the backlog, with higher positions indicating greater urgency.

- **Sprint goal alignment:** To measure sprint goal alignment, we assess the textual similarity between the content of each user story and sprint goal statement. For each user story, we concatenate the title and description into a single text document. For the sprint goal statement, we use the text as a separate document. We then pre-process these documents by converting the text to lowercase and removing punctuation and stop words. Using the Doc2Vec technique [45], we generate fixed-length vector representations for both the user stories and the sprint goal documents. To quantify the alignment, we calculate the cosine similarity between the embeddings of the user stories and the sprint goals.
- **Affinity:** To measure the relatedness between user stories, we use a procedure similar to the one used for assessing sprint goal alignment. Instead of calculating the similarity between the embeddings of user stories and sprint goals, we compute the cosine similarity between the Doc2Vec-generated embeddings of the user stories themselves. The resulting matrix of cosine similarity scores provides insights into the affinity among the stories in the backlog. To calculate the affinity score for a sprint, we sum the cosine similarity scores for all unique pairs of user stories selected for the sprint and normalize this sum by the number of pairs. For each pair of stories on the backlog, we multiply their similarity score by the product of their selection variables.
- **Delay risk:** To estimate delay risk, we follow a procedure outlined by Kula et al. [44] for predicting delay likelihood in user stories. Their method achieved an average F1 score of 76–84% across a large industrial dataset. We extract the 13 most significant predictor variables, identified as having an importance value higher than 0.05 in the study of Kula et al., and augment these with the Doc2Vec-generated story embeddings as an additional input feature. Since Kula et al. utilized the same backlog management tool, we were able to directly extract these variables from ING’s data using the same methodology. A detailed overview of the extracted variables is provided in the supplemental material [43]. We focus exclusively on user stories marked as ‘Completed’ in the backlog data, classifying a story as ‘delayed’ if it was postponed for one or more sprints. To simulate a realistic planning scenario, we extract predictor variables as they were recorded on the day of sprint planning for the sprint to which the story was originally assigned. For model building, we compare and evaluate four different classifiers that have been shown to be effective in risk prediction: Random Forests [8], AdaBoost [26], Multi-layer Perceptron [57] and Naive Bayes [12, 23]. A summary of the evaluation results can be found in the supplemental material [43]. A comparison shows that Random Forests outperforms the other classifiers. Therefore, we employ Random Forests and build predictive models tailored to each team, trained and tested on individual teams’ backlogs. We sort the user stories chronologically by their start dates, and use a 70-30 split for training and evaluation. The initial 70% of the stories are allocated to the training set, and the remaining 30% to the test set. This approach ensures that the model learns from historical data preceding the stories it is tested on.
- **Strategic alignment:** We were unable to measure or find proxies for strategic alignment. ING does not collect quantitative data on this factor nor has a standardized method for strategy reporting.

4.3 Predicting Story Selection Likelihood

We develop a method to learn story selection based on team planning behavior. Specifically, we build models that learn from a team’s sprint history to predict the likelihood of the team selecting a particular story for the upcoming sprint. These models identify the types of stories, or combinations of prioritization criteria, that teams select for their sprints under given constraints. We use *binary classification* and build team-specific models, training and testing them with historical backlog data from a specific team. For each sprint, we extract a historical snapshot of the backlog as recorded on the day of sprint planning, with the corresponding stories serving as input instances for the model. The number of user stories used for training each team-specific model ranges from 1,287 to 2,050. Input features include the *team velocity* set for the sprint, the estimated prioritization criteria for the stories, and the number of outgoing *dependencies* for each story on the backlog. Stories are labeled based on whether they were selected for the respective sprint.

We evaluated four machine learning algorithms suitable for classification tasks in software project management: Random Forests [8, 13], Multi-Layer Perceptron [4, 57], Least Median Square [4], and Naive Bayes [12, 23]. The results of our evaluation can be found in the supplemental material [43]. A comparison of the predictive performance demonstrated that Random Forests outperforms the other classifiers, with an average improvement of 6–18% in precision, 10–24% in recall, and 7–20% in F1 score. Therefore, we chose Random Forests for our experimental setup.

To simulate a real planning scenario, where decisions rely on insights from previous sprints, we sort the sprints chronologically by their start dates. For training and evaluation, we use a 70-30 split: the initial 70% of sprints are allocated to the training set, and the remaining 30% are used for the test set.

4.4 Obtaining Team Planning Objectives

We derive team planning objectives from the weights assigned to the prioritization criteria in the survey. To account for the absence of strategic alignment, we re-scale the weights of the remaining factors. We then convert the factors and their adjusted weights into a weighted sum, which represents the team’s planning objective. This objective reflects the factors in the proportion that teams aim to optimize when selecting user stories for the upcoming sprint, with higher weights indicating greater importance. For example, team *t48* provided the following weightings: 0.30 for urgency, 0.20 for sprint goal alignment, 0.20 for business value, 0.15 for delay risk, 0.10 for affinity, and 0.05 for strategic alignment. We re-scale and convert these weights into the following planning objective, which is to be optimized over a backlog of N user stories:

$$\begin{aligned} \text{Objective } t48 = \max \sum_{i=1}^N & 0.31 \cdot \text{urgency}_i + 0.21 \cdot \text{business value}_i \\ & + 0.21 \cdot \text{sprint goal alignment}_i + 0.11 \cdot \text{affinity}_i \\ & - 0.16 \cdot \text{delay risk}_i \end{aligned}$$

All factors, except delay risk, contribute positively to the value of user stories and should be maximized in the selection process for sprint planning. In contrast, delay risk should be minimized, which is why it is assigned a negative coefficient.

4.5 Optimization Model Development

We formalize sprint plan optimization using the 0-1 knapsack problem, where sprints are treated as knapsacks and user stories as items. The capacity of each sprint is defined by the team’s velocity, and each story’s weight is measured in story points. The objective is to select a subset of user stories that maximizes both the team’s planning objective and the estimated story selection likelihood. This subset must adhere to constraints related to the team’s velocity and story dependencies. We propose a linear programming model [19, 24] with the following variables:

- $u_i = 1$ if user story i is selected for the upcoming sprint, $u_i = 0$ otherwise;
- $\text{likelihood}(i)$ is the probability that story i will be selected by the team for the upcoming sprint, as predicted by our model using historical sprint data;
- velocity is the team’s capacity, i.e. the number of story points a team can deliver in a sprint;
- story points_i is the number of story points assigned to story i ;
- D_i is the set of user stories that story i depends on and that have not been completed yet;

The goal is to maximize the following objective function z by optimizing the assignment of the u_i variables over a backlog of N stories:

$$z = \max \sum_{i=1}^N \text{team planning objective} + \text{likelihood}(i) \quad (1)$$

subject to constraints:

$$\sum_{i=1}^N \text{story points}_i \cdot u_i \leq \text{velocity} \quad (2)$$

$$\sum_{j \in D_i} u_j \geq u_i \cdot |D_i| \quad (3)$$

The objective function z in Equation 1 aims to maximize the team planning objective and the selection likelihood estimate across the user stories on the backlog. Equation 2 constrains the total sum of story points for the selection of stories to not exceed the team’s velocity. The \geq symbol in Equation 3 ensures that all prerequisite stories for a given story are either completed beforehand or planned for the same sprint, thereby guaranteeing a logical sequence of story completion. We used the *Gurobi* solver in the Python library *PuLP* to solve the linear programming problem.

5 MODEL EVALUATION

Our evaluation aimed to answer the following research questions:

- **RQ2. Model alignment with team planning:** *Does the proposed approach generate sprint plans that align with teams’ actual sprint plans?* To evaluate how well the sprint plans generated by our model align with team planning, we compare the overlap in selected stories between our model-generated sprint plans and those created by the teams against a state-of-the-art (SoTA) baseline. We also assess the impact of different components of our objective function by analyzing the model’s performance using only the team planning objective, only the selection likelihood estimate, and the combined objective.

- **RQ3. Model effectiveness:** *How effective are the sprint plans generated by our model compared to teams’ actual sprint plans in terms of value delivery and risk mitigation?* We evaluate the effectiveness of our model-generated sprint plans against those created by the teams. This assessment involves aggregating the prioritization criteria across the user stories within each sprint plan to measure their overall effectiveness.
- **RQ4. Model usability:** *How do teams perceive the performance and usability of our model?* We apply the model to the current state of the backlogs and conduct interviews with teams to gather feedback and identify areas for improvement.

We address model alignment and effectiveness through a comparison with team’s actual sprint plans and a SoTA baseline [29, 30], and model usability through a qualitative evaluation at ING.

5.1 SoTA Baseline

We implement our optimization model using the single-sprint version of the objective function proposed by Golfarelli et al. [29, 30] to represent the SoTA baseline. This objective function aims to maximize the cumulative business value of the user stories selected for the sprint. The function is generic and does not account for team-specific contexts. A description of the objective function and our implementation can be found in the supplemental material [43]. The optimization model developed by Golfarelli et al. [29, 30] relies on team estimates for all prioritization criteria. However, obtaining retrospective team estimates for historical stories at ING was not feasible. Therefore, we use our procedure described in Section 4.2 to estimate the prioritization criteria, including delay risk as a proxy for the “uncertainty risk” factor defined by Golfarelli et al. [29, 30].

5.2 Experimental Setup

We perform experiments using the test set that comprises 30% of the teams’ past sprints (described in Section 4.3), incorporating our predictions of delay risk and selection likelihood for the user stories. To address RQ2, we compare the sprint plans generated by our model with those created by the teams in two ways. First, we assess the overlap of selected user stories using the *Jaccard Similarity* coefficient. This coefficient measures the proportion of user stories common to both our model’s selection and the teams’ selection, relative to the total number of unique user stories across both sets. A Jaccard Similarity value close to 1 indicates high similarity, while a value closer to 0 indicates lower similarity or greater dissimilarity. Second, we assess the semantic relatedness among user stories that differ between the model’s selection and the team’s selection. We compute cosine similarity scores for the Doc2Vec-generated embeddings [45] of these differing stories to measure semantic relatedness.

For RQ3, we evaluate the overall effectiveness of sprint plans by aggregating the prioritization criteria across the user stories in each sprint plan. To ensure comparability of urgency scores across different product backlogs, we normalize the urgency ranks by the total number of stories in each backlog, thus obtaining a relative urgency score. For performance comparison, we use the Wilcoxon Signed Rank Test [5] to determine the significance of the evaluation results. We measure the effect size using Vargha and Delaney’s \hat{A}_{12}

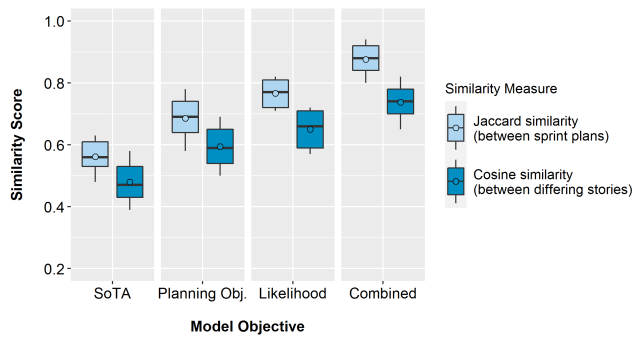


Figure 2: Comparison of story overlap (measured by Jaccard similarity; light blue) and semantic relatedness among differing stories (measured by cosine similarity; dark blue) between the teams' actual sprint plans and those generated by our model and the SoTA baseline. Results are shown for our model using the team planning objective (*planning obj.*) alone, the selection *likelihood* estimate alone, the *combined* approach, and the *SoTA* objective [29, 30].

statistic [5], a non-parametric measure commonly used in software project management [58].

5.3 (RQ2) Model Alignment

Figure 2 presents the evaluation results, comparing the story overlap (Jaccard similarity scores) and semantic relatedness among differing stories (cosine similarity scores) between the teams' actual sprint plans and those generated by our model and the SoTA baseline. On average, our model, using the combined objective function, achieves a high Jaccard similarity of 0.88 and a cosine similarity of 0.74. It significantly improves the models using either one of the individual objective components, with improvements of 15%–27% in Jaccard similarity and improvements of 13%–24% in cosine similarity. Statistical tests confirm significant improvements ($p < 0.001$) with medium to large effect sizes, ranging from 0.64 to 0.81. The model using the SoTA objective achieves the lowest scores, with an average Jaccard similarity of 0.56 and a cosine similarity of 0.48. All three versions of our model outperform the SoTA with large effect sizes greater than 0.81.

The proposed approach, integrating teams' planning objectives and behavior, is effective in generating sprint plans that are aligned with teams' actual sprint plans.

5.4 (RQ3) Model Effectiveness

Figure 3 shows the distributions of aggregated prioritization criteria across sprint plans generated by our model and those created by teams. The model-generated plans demonstrate notable improvements in effectiveness compared to the teams' plans, with higher cumulative business value, stronger sprint goal alignment, and more effective delay risk mitigation. On average, the model increases business value by 29%, improves sprint goal alignment by 14%, and reduces delay risk by 42%. Statistical tests confirm the significance of these improvements ($p < 0.001$), with medium to large effect sizes ranging from 0.66 to 0.87. Additionally, our model achieves an average improvement of 5% in affinity, which is

significant ($p < 0.01$) but with a small effect size ($\hat{A}_{12} = 0.57$). The differences in normalized urgency rank scores are not significant.

Our model drives improvements in team performance by generating sprint plans that deliver more business value, align more closely with sprint goals, and better mitigate delay risks.

5.5 (RQ4) Model Usability

5.5.1 Interview Methodology. The main goal of our qualitative analysis was to assess how teams perceive the performance and usability of our model. We conducted semi-structured interviews with 10 teams at ING to gather feedback and identify areas for improvement. We opted for a semi-structured format due to its flexibility in discussing prepared questions and exploring emergent topics [36]. Table 3 provides an overview of our interview questions. We primarily asked open-ended questions to encourage in-depth discussion without bias, as well as a focused question to examine the teams' willingness to use the model in practice. The study design was approved by the ethical review board of ING. We randomly selected and invited 10 teams from the pool of survey respondents to participate. We sent email invitations to the product owners of these teams, outlining the study's purpose and the required commitment. All teams accepted the invitation. Demographic details of the participating teams can be found in the supplemental material [43].

The interviews were conducted face-to-face by the first author at the start of the teams' sprint planning meetings. They lasted, on average, 41 minutes. Each interview began with a brief introduction to the study and a demonstration of the team's backlog. We applied our model to the current state of the backlog and presented the generated plan for the upcoming sprint in a ServiceNow mock-up. This sprint plan served as a focal point for group discussion on model performance and usability.

We recorded and transcribed the interviews for analysis. We used *open coding* [21, 60] to analyze the transcripts and summarize the responses, coding by statement and allowing codes to emerge throughout the process. We constantly compared and refined the codes, grouping similar ones into categories.

Table 3: Overview of Interview Questions

- **RQ4.1 Model alignment with team planning:** How well does the generated sprint plan align with your team goals and criteria for selecting stories?
- **RQ4.2 Model effectiveness:** Are there any modifications you would suggest for the generated sprint plan? If so, what changes would you propose and why?
- **RQ4.3 Model impact:** How do you foresee this model influencing your sprint planning?
- **RQ4.4 Model usability:** Would you use this model in practice? If so, how would you integrate it into your sprint planning process?

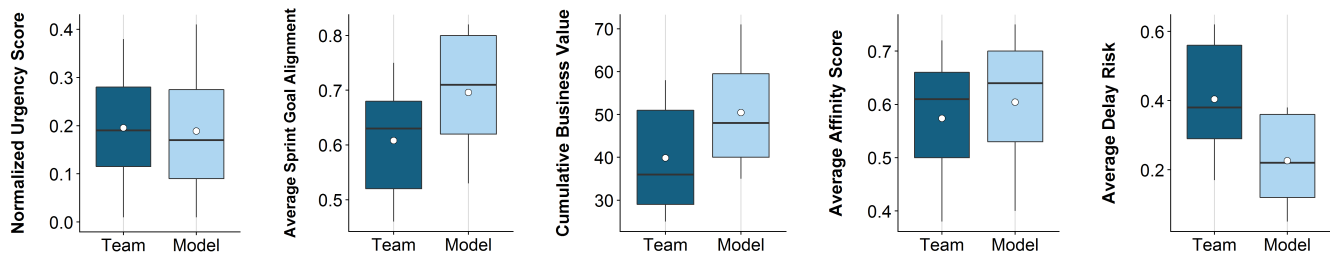


Figure 3: Comparison of aggregated prioritization criteria between *team*-composed and *model*-generated sprint plans. The *Normalized urgency score* reflects each story’s backlog position divided by the total number of stories, with lower scores indicating greater urgency and more effective planning. *Cumulative business value* is the sum of business value scores assigned to stories in a sprint. *Average sprint goal alignment* and *Average affinity score* refer to the cosine similarity scores between the embeddings of user stories and sprint goals, averaged over stories in the sprint plans. *Average delay risk* represents the average probability of story delay, with lower values indicating more effective planning.

5.5.2 Interview Results. This section presents the findings from our interviews. We use a [TX] notation to mark example quotes from the interviews, where ‘X’ refers to the identification number of the corresponding team. The codes resulting from our manual coding process are underlined.

RQ4.1: Model alignment with team planning. Consistent with our findings for RQ2, which showed a high Jaccard similarity score of 0.88, a majority of teams (seven out of ten) found that the model’s story selections aligned with their reasoning and intuition. Teams described the model as making “appropriate choices” [t04] and “intuitive selections” [t09] that match their objectives and sprint priorities. For instance, one team stated: “The proposed sprint plan primarily includes user stories that provide genuine value to the customer and that we would like to pick up in the next sprint.” [t04] Another team noted that the model effectively captures a balanced value-risk trade-off: “The model picked user stories that are valuable or urgent to the customer and ready for implementation. A few user stories at the top of our backlog still require further refinement before implementation can begin. The model seems to address that properly.” [t08] However, two teams had reservations regarding the implementation order of user stories. One team commented: “While this [sprint] plan is very much in line with what our customers want right now, it contains several high-priority stories that we would normally divide across multiple sprints to dedicate sufficient attention to each.” [t10] Another team perceived a lack of long-term perspective regarding the product’s trajectory: “The model does not differentiate between customer value and product value. It prioritizes customer needs but misses out on the bigger picture of where the product should be going.” [t05] One team noted that their changing objectives, resulting from strategic shifts in their project, were not reflected in the generated sprint plan: “Due to recent budget shifts, we have shifted our priority to smaller and low-risk user stories, differing from what we reported in the survey. The model’s selections reflect our goals from the past year but do not align with our current situation.” [t03]

RQ4.2: Model effectiveness. Consistent with our findings for RQ3, which showed increased effectiveness in the sprint plans generated by our model, most teams (six out of ten) indicated they would make only minor tweaks or refinements to the sprint plan rather than substantial alterations. Teams’ suggestions regarding

potential modifications revealed several themes. Four teams highlighted the need to postpone selected stories due to changes in team composition. For example, one team explained: “Next sprint one of our senior members will be absent. While the model adapted by creating a smaller sprint, it selected two stories that rely on the expertise of the senior member. Our typical course of action is to postpone these stories until the required developer returns.” [t05] Three teams preferred to postpone selected stories that require further refinement and are not yet ready for implementation. For instance, one team mentioned: “The proposed plan includes three stories for which we currently lack a clear approach. We still need to do some research and break these stories down into smaller tasks to gain a better understanding of the required solution.” [t01] Other mentioned modifications included adjusting the order of user stories, such as advancing risky stories to early sprints to avoid late side-effects.

RQ4.3: Model impact. Teams identified three key benefits of integrating our model into their sprint planning process. Firstly, they noted that the model could improve productivity by facilitating quicker decision-making. For example, one team stated: “The model guides teams to focus their discussions on the inclusion of pre-selected stories, reducing the need for lengthy, exploratory conversations typical of creating a sprint from scratch.” [t02] Secondly, teams believed that the model could enhance business alignment in sprint planning. A team member explained: “The model could help us focus on customer needs rather than internal interests.” [t06] Lastly, teams suggested that the model could facilitate informed decision-making by “providing insights into the backlog” [t01] and stimulating discussions on overlooked stories. One team stated: “We discussed stories that had been lingering in the backlog for a while and were typically skipped during our meetings.” [t04] However, some teams also mentioned potential drawbacks of the model, such as overreliance on the model and reduced communication among team members. One team explained: “Over time, teams might become dependent on the model, resulting in decreased communication within the team and limited exploration of alternative approaches.” [t06]

RQ4.4: Model usability. Eight teams expressed their willingness to use the model as part of their sprint planning meetings. They see the model as a support for human judgement, to be used in conjunction with existing planning strategies, rather than as a

replacement. The teams emphasized the importance of maintaining control over the sprint plan and having the flexibility to adjust it as needed. They favored an interactive format that allows them to modify the sprint and team objectives on the go. Some teams stressed the importance of model explainability, suggesting that the model would be more useful if it provided explicit rationale for story selection. For instance, one team explained: “*We would like to understand why the model selects this particular set of user stories and how it affects the feasibility of the sprint if we decide to choose other stories.*” [t07] The two teams that were hesitant to use the model in practice expressed concerns about overreliance on it, fearing it could diminish the quality of team discussions.

6 DISCUSSION

6.1 Recommendations for Practitioners

Improving sprint planning. Our model improves the efficiency and outcomes of sprint planning. We identified a set of key factors and project context variables that influence story prioritization. By incorporating these factors into an optimization model, teams can generate context-aware sprint plans that align with their goals and performance. Using a combination of team-supplied and data-derived weights, our model achieves an 88% overlap with team-selected stories and 74% semantic relatedness with differing choices. This indicates that the model effectively captures team priorities. In cases where the model diverges, it often makes better decisions, improving value-risk balance and project outcomes. Our interview findings further confirm the model’s effectiveness and alignment with team planning.

In terms of speed, the time required to generate a sprint plan depends on the number of user stories and dependencies on the backlog. With precomputed outputs of the trained predictive models, evaluations on an Intel Core i9 with 32GB RAM at 5.8 GHz showed that for 71% of teams (with fewer than 100 stories), computation time ranged from seconds to 2 minutes; for larger backlogs (over 100 stories), computation time ranged from 2 to 6 minutes. This is a significant improvement compared to sprint planning meetings, which typically take hours.

Overall, our model streamlines sprint planning, empowering teams to achieve greater productivity and better project outcomes. This is further confirmed by the majority of teams expressing willingness to use the model in practice, citing improvements in productivity, business alignment, and more informed decision-making.

Interactive support tool. While our model offers substantial benefits, it is designed to support, rather than replace, human judgement. Our interview results indicate that teams prefer to integrate the model with their existing planning methods to maintain control over the process. We recommend an interactive approach that allows teams to adjust objectives and proposed plans as needed. An interactive format also enables the consideration of team composition, availability, and story readiness by gathering team input through forms or backlog tool updates. This flexibility ensures that the model remains aligned with evolving objectives and mitigates concerns about overreliance. Previous research [4, 22] has explored interactive methods for incorporating human expertise into release plan optimization.

Model explainability. Our qualitative evaluation highlights the importance of providing a rationale for story selection to improve model usability. Techniques such as Scenario Discovery [28, 46] and post-optimal analysis (e.g., [31, 40]) can be used to examine how variations in input parameters or assumptions impact the outcomes of an optimization model. This analytical process can help teams uncover patterns among variables and better understand the model’s story selection. For example, teams may discover that certain types of stories are prioritized in response to specific project constraints or that story priorities shift based on project urgency.

Implementation effort. Our model learns from past team performance to estimate delay risk and identify patterns in planning behavior, making its insights team-specific. To address data scarcity, especially with new teams [47], we recommend developing a generalized model trained at the product or department level. This approach involves training on historical log data from multiple teams within a product or department, while still allowing individual teams to supply factor weights for model customization. Further analysis at ING shows that product- and department-level models achieve moderate Jaccard similarities (0.62 to 0.68) and cosine similarities (0.59 to 0.63) between proposed and actual sprint plans. These results suggest that generalized models can provide a reasonable baseline for new teams until sufficient team-specific data is available.

6.2 Implications for Researchers

Importance of prioritization criteria. Agile methods typically emphasize business value as the main prioritization criterion, a topic that has sparked debate in prior research [51, 54]. Our survey results reveal a more nuanced perspective. We found that the priority of a story is determined by a combination of factors, with urgency, sprint goal alignment, and business value as key drivers. However, the impact of these factors varies across project settings, highlighting the need for contextual support in sprint planning. Future work should examine the influence of project characteristics on prioritization criteria through statistical controls, such as multiple regression analysis. A deeper understanding could inform the redesign or reframing of agile prioritization methods to better align with actual practice.

Strategic alignment and delay risk. In our study, strategic alignment emerged as a new factor affecting story prioritization. Although it ranked lower in survey responses, interview findings emphasize the importance of integrating long-term product strategy into sprint plan generation. While not currently integrated into our model, measuring strategic alignment holds potential for future research. One approach could involve reviewing strategy documents and comparing them with story descriptions to assess alignment. Additionally, our analysis of delay risk highlights its significant impact on story selection for sprint planning. While our current approach estimates delay risk for individual stories, future work could estimate overall delay risk for the sprint plan by considering dependencies among stories and propagation effects. Initial efforts in this direction have been made by Choetkiertikul et al. [11] using network analysis.

Scenario-based modeling. Interview results suggest that using complementary techniques, such as scenario-based modeling, could

improve the model's alignment with team planning. Teams frequently adjust their prioritization of user stories in response to events, such as delays or strategic shifts. Future research should focus on identifying scenarios that impact sprint planning, either through direct team input or automated extraction from backlog data. Prior studies by Sutcliffe et al. [63] and Regnell et al. [56] have explored scenario-based modeling for requirements engineering.

7 THREATS TO VALIDITY

Internal validity. We acknowledge that surveys and interviews may contain ambiguous questions and introduce biases [48]. To address this, we used terminology familiar to the case company, ordered questions sequentially, and randomized the order of prioritization criteria [62]. To counter social desirability bias [27], we informed participants that the responses would be kept confidential. Our survey may have been subject to non-response bias [20], especially among teams struggling to meet sprint commitments.

Construct validity. Poor record keeping may have influenced the data variables we used to measure prioritization criteria and story delay [55]. Story delay is assessed based on the number of sprints a story has been part of, yet inaccuracies may occur if teams close their stories too early or too late. Although ING encourages teams to deliver on-time, some teams may not take their sprint commitments seriously, adding stories to sprints without intent to deliver. We addressed these concerns by collecting data from a large number of sprints and teams over a four year span.

External validity. The generalizability of findings is an important concern for any single-case study. Although we analyzed data from various teams and products, our findings may not be representative of software projects in other organizations or open-source settings. In other settings, teams may have more dynamic setups and different planning practices. While the high number of teams and availability of historical data may be more common in large-scale organizations, we expect our findings on the prioritization criteria and model impact to be transferable to other settings, regardless of scale. Our approach to generating sprint plans can be applied as is to backlog data from other agile software organizations. It is important to note that ING's strict security regulations as a financial organization may have influenced the prioritization criteria rankings. As a result, these rankings may be more relevant to organizations with similar business- or safety-critical systems. Further research is required to validate our findings in other settings and reach more general conclusions.

8 CONCLUSIONS

Sprint planning is crucial for the successful execution of agile software projects. While various prioritization criteria influence the selection of user stories for sprint planning, their relative importance remains largely unexplored, especially across diverse project contexts. In this paper, we investigated how prioritization criteria vary across project settings and proposed a context-aware optimization model to generate sprint plans that align with team goals and performance. By integrating teams' planning objectives and sprint history, the model adapts to team contexts, estimating prioritization criteria and identifying patterns in planning behavior. We applied our approach to real-world data from thousands of sprints

and evaluated our model through both quantitative and qualitative analyses. The key findings of this study include:

- (1) Urgency, sprint goal alignment, and business value emerged as the most important prioritization criteria. Their influence varies depending on project characteristics, such as resources, priority, client type, and security level.
- (2) Our model outperforms the state-of-the-art in aligning with team planning and boosts team performance by generating sprint plans that deliver more business value, align more closely with sprint goals, and better mitigate delay risks.
- (3) The majority of teams found our approach to be consistent with their goals and valuable as interactive support.

We identified promising areas for future research, including interactive optimization, scenario-based modeling, and model explainability. Progress in these areas is crucial to better understand and support planning practices in agile software development.

REFERENCES

- [1] ACHIMUGU, P., SELAMAT, A., IBRAHIM, R., AND MAHRIN, M. N. A systematic literature review of software requirements prioritization research. *Information and software technology* 56, 6 (2014), 568–585.
- [2] AL-TA'ANI, R. H., AND RAZALI, R. Prioritizing requirements in agile development: A conceptual framework. *Procedia Technology* 11 (2013), 733–739.
- [3] AL-ZUBAIDI, W. H. A., DAM, H. K., CHOETKIERTIKUL, M., AND GHOSE, A. Multi-objective iteration planning in agile development. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)* (2018), IEEE, pp. 484–493.
- [4] ARAÚJO, A. A., PAIXAO, M., YELTSIN, I., DANTAS, A., AND SOUZA, J. An architecture based on interactive optimization and machine learning applied to the next release problem. *Automated Software Engineering* 24 (2017), 623–671.
- [5] ARCURI, A., AND BRIAND, L. A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. *Software Testing, Verification and Reliability* 24, 3 (2014), 219–250.
- [6] BLOCH, M., BLUMBERG, S., AND LAARTZ, J. Delivering large-scale it projects on time, on budget, and on value. *Harvard Business Review* 5, 1 (2012), 2–7.
- [7] BOSCHETTI, M. A., GOLFARELLI, M., RIZZI, S., AND TURRICCHIA, E. A lagrangian heuristic for sprint planning in agile software development. *Computers & Operations Research* 43 (2014), 116–128.
- [8] BREIMAN, L. Random forests. *Machine learning* 45, 1 (2001), 5–32.
- [9] CAO, L., AND RAMESH, B. Agile requirements engineering practices: An empirical study. *IEEE software* 25, 1 (2008), 60–67.
- [10] CERVONE, H. F. Understanding agile project management methods using scrum. *OCLC Systems & Services: International digital library perspectives* (2011).
- [11] CHOETKIERTIKUL, M., DAM, H. K., TRAN, T., AND GHOSE, A. Predicting delays in software projects using networked classification (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (2015), IEEE, pp. 353–364.
- [12] CHOETKIERTIKUL, M., DAM, H. K., TRAN, T., AND GHOSE, A. Predicting the delay of issues with due dates in software projects. *Empirical Software Engineering* 22 (2017), 1223–1263.
- [13] CHOETKIERTIKUL, M., DAM, H. K., TRAN, T., PHAM, T., GHOSE, A., AND MENZIES, T. A deep learning model for estimating story points. *IEEE Transactions on Software Engineering* 45, 7 (2018), 637–656.
- [14] CHOW, T., AND CAO, D.-B. A survey study of critical success factors in agile software projects. *Journal of systems and software* 81, 6 (2008), 961–971.
- [15] COCKBURN, A., AND HIGHSMITH, J. Agile software development, the people factor. *Computer* 34, 11 (2001), 131–133.
- [16] COHN, M. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [17] COHN, M. *Agile estimating and planning*. Pearson Education, 2005.
- [18] DAM, H. K., TRAN, T., GRUNDY, J., GHOSE, A., AND KAMEI, Y. Towards effective ai-powered agile project management. In *2019 IEEE/ACM 41st international conference on software engineering: new ideas and emerging results (ICSE-NIER)* (2019), IEEE, pp. 41–44.
- [19] DANTZIG, G. B. Linear programming. *Operations research* 50, 1 (2002), 42–47.
- [20] DE LEEUW, E. D. *Data quality in mail, telephone and face to face surveys*. ERIC, 1992.
- [21] DEFranco, J. F., AND LAPLANTE, P. A. A content analysis process for qualitative software engineering research. *Innovations in Systems and Software Engineering* 13, 2 (2017), 129–141.

- [22] DO NASCIMENTO FERREIRA, T., ARAÚJO, A. A., NETO, A. D. B., AND DE SOUZA, J. T. Incorporating user preferences in ant colony optimization for the next release problem. *Applied Soft Computing* 49 (2016), 1283–1296.
- [23] DOMINGOS, P., AND PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning* 29, 2 (1997), 103–130.
- [24] DORFMAN, R., SAMUELSON, P. A., AND SOLOW, R. M. *Linear programming and economic analysis*. Courier Corporation, 1987.
- [25] DRURY-GROGAN, M. L. Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors. *Information and software technology* 56, 5 (2014), 506–515.
- [26] FREUND, Y., AND SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.
- [27] FURNHAM, A. Response bias, social desirability and dissimulation. *Personality and individual differences* 7, 3 (1986), 385–400.
- [28] GOERIGK, M., AND HARTISCH, M. A framework for inherently interpretable optimization models. *European Journal of Operational Research* 310, 3 (2023), 1312–1324.
- [29] GOLFARELLI, M., RIZZI, S., AND TURRICCHIA, E. Sprint planning optimization in agile data warehouse design. In *Data Warehousing and Knowledge Discovery: 14th International Conference, DaWaK 2012, Vienna, Austria, September 3–6, 2012. Proceedings 14* (2012), Springer, pp. 30–41.
- [30] GOLFARELLI, M., RIZZI, S., AND TURRICCHIA, E. Multi-sprint planning and smooth replanning: An optimization model. *Journal of systems and software* 86, 9 (2013), 2357–2370.
- [31] GREENBERG, H. J. The use of the optimal partition in a linear programming solution for postoptimal analysis. *Operations Research Letters* 15, 4 (1994), 179–185.
- [32] GRENNING, J. Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting* 3 (2002), 22–23.
- [33] HARRIS, R. S., AND COHN, M. Incorporating learning and expected cost of change in prioritizing features on agile projects. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (2006), Springer, pp. 175–180.
- [34] HODA, R., SALLEH, N., AND GRUNDY, J. The rise and evolution of agile software development. *IEEE software* 35, 5 (2018), 58–63.
- [35] HOLM, S. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.
- [36] HOVE, S. E., AND ANDA, B. Experiences from conducting semi-structured interviews in empirical software engineering research. In *11th IEEE International Software Metrics Symposium (METRICS'05)* (2005), IEEE, pp. 10–pp.
- [37] JANZI, S., AND RAJESWARI, K. A greedy heuristic approach for sprint planning in agile software development. *International Journal for Trends in Engineering & Technology* 3, 1 (2015), 18–21.
- [38] KASUNIC, M. Designing an effective survey. Tech. rep., Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2005.
- [39] KITCHENHAM, B. A., AND PFLIEGER, S. L. Principles of survey research: parts 1 – 6. *ACM SIGSOFT Software Engineering Notes* 26–28 (2001 - 2003).
- [40] KLEIN, D., AND HOLM, S. Integer programming post-optimal analysis with cutting planes. *Management Science* 25, 1 (1979), 64–72.
- [41] KNIBERG, H., AND IVARSSON, A. Scaling agile@ spotify with tribes, squads, chapters & guilds. *Entry posted November 12* (2012).
- [42] KONTIO, J., HOGELUND, M., RYDEN, J., AND ABRAHAMSSON, P. Managing commitments and risks: challenges in distributed agile development. In *Proceedings. 26th International Conference on Software Engineering* (2004), IEEE, pp. 732–733.
- [43] KULA, E., VAN DEURSEN, A., AND GEORGIOS, G. Supplemental material for Context-Aware Automated Sprint Plan Generation for Agile Software Development, year = 2024, url = <https://doi.org/10.5281/zenodo.11522834>.
- [44] KULA, E., VAN DEURSEN, A., AND GOUSIOS, G. Modeling team dynamics for the characterization and prediction of delays in user stories. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (2021), IEEE, pp. 991–1002.
- [45] LE, Q., AND MIKOLOV, T. Distributed representations of sentences and documents. In *International conference on machine learning* (2014), PMLR, pp. 1188–1196.
- [46] LEMPERT, R. J., BRYANT, B. P., AND BANKES, S. C. Comparing algorithms for scenario discovery. *RAND, Santa Monica, CA* (2008).
- [47] LIKA, B., KOLOMVATOS, K., AND HADJIEFTHYMIADIS, S. Facing the cold start problem in recommender systems. *Expert systems with applications* 41, 4 (2014), 2065–2073.
- [48] MOLLÉRI, J. S., PETERSEN, K., AND MENDES, E. An empirically evaluated checklist for surveys in software engineering. *Information and Software Technology* 119 (2020), 106240.
- [49] MORAN, A. Agile risk management. In *Agile Risk Management*. Springer, 2014, pp. 33–60.
- [50] OZCELIKKAN, N., TUZKAYA, G., ALABAS-USLU, C., AND SENNAROGLU, B. A multi-objective agile project planning model and a comparative meta-heuristic approach. *Information and Software Technology* 151 (2022), 107023.
- [51] PETERSEN, K., AND WOHLIN, C. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software* 82, 9 (2009), 1479–1490.
- [52] PIKKARAINEN, M., HAIKARA, J., SALO, O., ABRAHAMSSON, P., AND STILL, J. The impact of agile practices on communication in software development. *Empirical Software Engineering* 13 (2008), 303–337.
- [53] POPLI, R., AND CHAUHAN, N. Agile estimation using people and project related factors. In *2014 International Conference on Computing for Sustainable Global Development (INDIACom)* (2014), IEEE, pp. 564–569.
- [54] RACHEVA, Z., DANEVA, M., SIKKEL, K., HERRMANN, A., AND WIERINGA, R. Do we know enough about requirements prioritization in agile projects: insights from a case study. In *2010 18th IEEE International Requirements Engineering Conference* (2010), IEEE, pp. 147–156.
- [55] RALPH, P., AND TEMPERO, E. Construct validity in software engineering research and software metrics. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (2018), pp. 13–23.
- [56] REGNEL, B., RUNESON, P., AND THELIN, T. Are the perspectives really different?—further experimentation on scenario-based reading of requirements. *Empirical Software Engineering* 5 (2000), 331–356.
- [57] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [58] SARRO, F., PETROZZIELLO, A., AND HARMAN, M. Multi-objective software effort estimation. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)* (2016), IEEE, pp. 619–630.
- [59] SCHWABER, K., AND BEEDLE, M. *Agile software development with Scrum*, vol. 1. Prentice Hall Upper Saddle River, 2002.
- [60] SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering* 25, 4 (1999), 557–572.
- [61] SPEARMAN, C. The proof and measurement of association between two things.
- [62] STRACK, F. “order effects” in survey research: Activation and information functions of preceding questions. In *Context effects in social and psychological research*. Springer, 1992, pp. 23–34.
- [63] SUTCLIFFE, A. G., MAIDEN, N. A., MINOCHA, S., AND MANUEL, D. Supporting scenario-based requirements engineering. *IEEE Transactions on software engineering* 24, 12 (1998), 1072–1088.
- [64] TAVARES, B. G., DA SILVA, C. E. S., AND DE SOUZA, A. D. Practices to improve risk management in agile projects. *International Journal of Software Engineering and Knowledge Engineering* 29, 03 (2019), 381–399.
- [65] USMAN, M., MENDES, E., AND BÖRSTLER, J. Effort estimation in agile software development: a survey on the state of the practice. In *Proceedings of the 19th international conference on Evaluation and Assessment in Software Engineering* (2015), pp. 1–10.
- [66] USMAN, M., MENDES, E., WEIDT, F., AND BRITTO, R. Effort estimation in agile software development: a systematic literature review. In *Proceedings of the 10th international conference on predictive models in software engineering* (2014), ACM, pp. 82–91.