May 2023

MAPM: MULTI-MODAL AUTO-AGILE PROJECT RISK MANAGEMENT AND PREDICTION FOR COLLABORATION PLATFORM USING AI/ML

Pengfei Sun

Qihong Shao

MAPM: MULTI-MODAL AUTO-AGILE PROJECT RISK MANAGEMENT AND PREDICTION FOR COLLABORATION PLATFORM USING AI/ML

AUTHORS:
Pengfei Sun
Qihong Shao

ABSTRACT

Agile software development tools have been created to assist project management and enhance productivity. However, it may be challenging to properly employ those tools, especially in a hybrid work environment. Techniques are presented herein, which may be referred to herein as a MaPM system, that leverage a conferencing platform to offer a real-time agile project risk management and prediction framework that utilizes multi-modal collaboration data sources. Aspects of the presented techniques encompass an artificial intelligence (AI)-backed summarization model that may be utilized to extract project details and an auto-agile that model may consume those loggings and generate the predicted project sprint backlogs and their risks. Further aspects of the presented techniques support an optimization module that may jointly update the predicted sprint backlogs and the estimated task risks to realize the finalized backlog sequences. In summary, an MaPM system, according to the presented techniques, offers four novelties compared with conventional agile project management tools – a conferencing platform-centralized solution for automatic agile project risk prediction and management, a real-time multi-modal-based project risk monitoring and prediction system, the generation of sprint backlogs based on fully evaluated contexts that are collected from all of a project's participants, and the liberation of project contributors from having to conduct manual project tracking and recording.

DETAILED DESCRIPTION

Agile software development tools have been created to assist project management and enhance productivity. However, a persistent challenge with such tools, especially in a hybrid work environment, is that people often find it exhausting to manually document the progress of a project and track the rapid changes in the same by navigating among different tools. Additionally, numerous collaborations and communications regarding variabilities

Published by Technical Disclosure Commons, 2023

can potentially delay different tasks and cause a project to overspend. Further, to establish panorama maneuvering, summarizing and logging the daily communication among multiple collaborators can be quite time consuming.

As a result, a project management approach that relies on conventional agile tools has three potential negative impacts on productivity. A first potential impact encompasses the challenge of manually documenting and tracking progress. In an agile project management system, contributors pre-schedule or submit their progress individually to the platform. Task owners then need to frequently check various sources to collect updates and adapt to requests in a timely manner. However, such a distributed approach to tracking progress can be time consuming and inefficient, particularly within a fast-paced agile project in a hybrid work environment.

A second potential impact encompasses a lack of full modality as-a-context monitoring. Conventional agile tools rely on diligent inputs from contributors to consolidate task management. However, people often struggle to consider all of the context of progress and tasks, as the collaborations are scattered across multiple modalities such as video conferencing, instant messages, code snippets, and short conversations. As a result, decisions and strategies based on agile sprint backlogs are at a higher risk on conventional agile platforms as they do not capture all of the modalities within a collaborative environment. Such an approach can potentially lead to gaps between aligning the sprint backlog and the project risk and needs since no full picture of a project can be obtained based on the conventional agile tools.

A third potential impact reflects the fact that heuristics-based project risk management is less effective. The failure of a project can be caused by various factors, including a lack of awareness of setbacks among collaborators' work, delays without timely notifications and negotiation among contributors, and conflicts between different priorities. Collaborators may hold several meetings to promptly address these issues, but conventional agile tools may not capture such risks in real time, relying solely on observations that are summarized by project managers and team members. As a result, a heuristics-based risk management approach may be less effective in a real agile project as it is more vulnerable to oversight.

Recently, there has been a growing trend of incorporating machine learning-driven methodologies into project management practices, particularly in the context of modern agile projects. For example, in one approach a framework for general artificial intelligence (AI)-based agile project management was proposed. Other studies have focused on developing analytic engines and reasoning capabilities for AI-governed project management platforms. In another approach, machine learning strategies were also used to strengthen the project scope definition based on an agile environment. However, the above-described approaches primarily emphasize building user-isolated conventional platforms that can be augmented with AI technologies to enable the automatic analysis of project logs.

As a result, despite the inclusion of AI engines those user-isolated platforms are unable to offer a real-time project management system. In order to input multi-modal project data into those platforms for AI-driven analysis, contributors still need to manually sift through large amounts of collaboration data, identify relevant pieces, and then pass those pieces on to an AI engine to generate recommendations. Furthermore, such an AI-backed analysis engine operates on resources that are filtered by each individual contributor. However, the process of selecting and filtering the original source data can introduce biases, potentially leading to incomplete contexts and, consequently, an effect on the accuracy of risk predictions.

To address the challenges that were described above, techniques are presented herein that leverage a multi-modal agile project management system to automatically generate sprint backlogs by utilizing a conferencing platform. Figure 1, below, presents elements of such an approach.

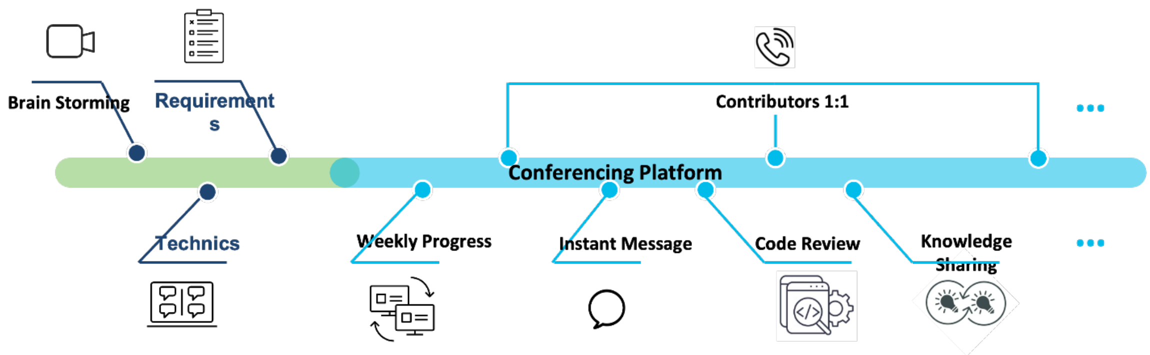

Figure 1: Multi-modal Conferencing Platform in Agile Project Lifecycle

As illustrated in Figure 1, above, a conferencing platform may, particularly in today's hybrid work environment, work as a centralized interface that supports the entire lifecycle of an agile project. Collaborators may share detailed updates and information through the conferencing platform, ranging from idea formalization to final product delivery and project documentation. Further, project managers may gather progress updates on each subtask and engage in internal discussions to solidify future sprint backlogs. In the case of setbacks or subtask conflicts, colleagues may share their opinions and intensively brainstorm on the conferencing platform to solicit solutions.

Considering that the progress updates for an agile project are in various formats (such as a video stream, an audio clip, text, slides, code snippets, and screenshots), a multi-modal summarization model may serve as an effective tool to extract the most relevant information. Summarized item lists may be further fed into an auto-agile model (as described and illustrated below) to generate a backlog sequence along with the risk estimations for each task. The detailed list that is generated from an encoder-decoder transformer model can capture the conflicts, delays, resource shortages, setbacks, and other failure factors. In addition, sequence optimization may be leveraged to update the detailed list jointly with the estimated risk scores. The finalized sprint backlogs may be wrapped as the sprint for project managers to consume.

Throughout a project's lifecycle, individual contributors will make progress and communicate technical details using various modalities, such as a video stream, a whiteboard session, documentation, code snippets, instant messages, and meeting minutes.

By combining these multi-modal project loggings, an underlying intelligent machine learning model can enable automated agile project tracking and management.

As introduced above, and as will be described and illustrated below, the techniques presented herein (which for simplicity of exposition may be referred to herein as a MaPM system) encompass four major elements. A first element leverages different conferencing platforms as a centralized hub to generate a multi-modal automatic agile project management system. A second element comprises an AI-driven project management system that is integrated within a conferencing platform to provide for multi-threaded, real-time project tracking and monitoring across all of the different modalities. A third element comprises the integration of multiple modalities to strengthen the adaptability in risk management. And finally, under a fourth element the use of an AI-driven agile project management approach enables the system to continually learn from data and consistently improve project risks management.

The next section of the instant narrative describes the architecture of a MaPM system according to the techniques presented herein.

Agile project management focuses on rapid delivery and the embracing of changes. The sprint backlogs need to be adapted to various scenarios such as team members being in and out; compromising on priorities; subtask delays, setback, and migration; etc. The sprint backlog may be continually updated and refined to ensure that it contains items that are relevant to the project's scope and objectives, with sufficient detail and appropriate estimation. Each sprint is a short period during which the team aims to complete a subset of items from the sprint backlog. Prior to a sprint, the team conducts sprint planning to identify the goal of the upcoming sprint and then selects items from the product backlog that they will work on to achieve the sprint's goal.

Figure 2, below, depicts elements of an exemplary MaPM architecture according to the techniques presented herein and reflective of the above discussion.

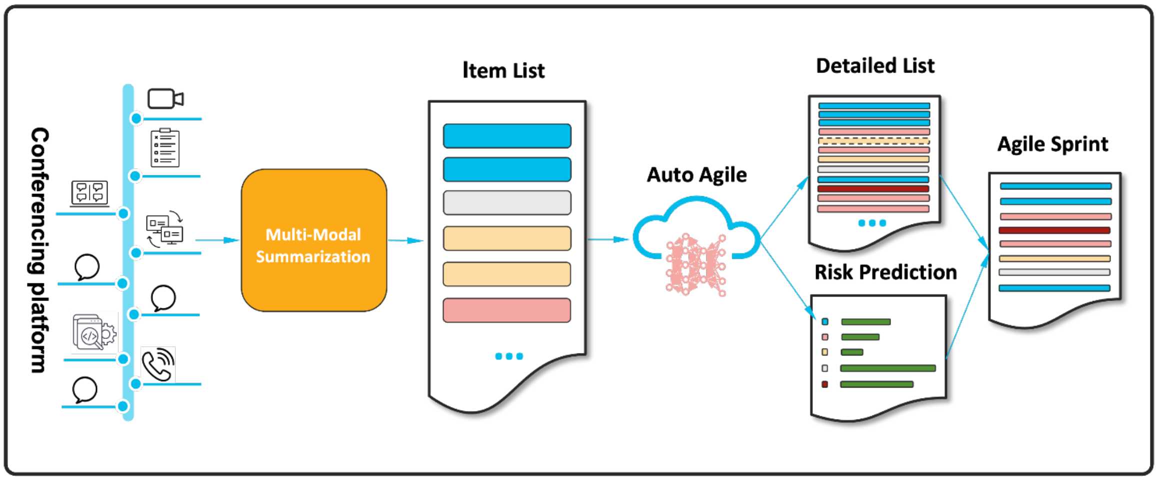

Figure 2: Conferencing Platform-based AI-driven Multi-modal Project Management

As shown in Figure 2, above, the exemplary architecture shows how an AI backend model may be utilized, based on a conferencing platform, to scale up a current project management approach. Unlike the conventional agile tools for managing the backlogs of a project, a conferencing platform-based agile project management system according to the techniques presented herein empowers all of the team contributors to prioritize, update, and monitor a project's progress by employing the real-time, multi-modal collaborations.

A MaPM architecture, according to the techniques presented herein, includes three key elements, each of which will be briefly introduced below and then described and illustrated later in the instant narrative.

A first element encompasses a multi-modal summarization module that may transform the collaboration data into an item list across all of the project contributors in the time domain. The generated item list may include high-level descriptions of the communications among the different collaborators. Specifically, all of the project progress from the different contributors may be recorded with respect to challenges, setbacks, pain points, prioritizations, etc.

A second element encompasses an encoder-decoder based auto-agile module that may be leveraged to further transform an item list into a detailed list along with the predicted risks for each pair.

A third element encompasses a sprint backlog optimization module that implements a series of operations (such as insertion, shuffling, splitting, and deletion) to optimize the task lists and reduce the project risk in each sprint.

As noted above, a first key element of a MaPM solution (according to the techniques presented herein) encompasses a multi-modal project summarization module.

The different collaborations within a project-oriented conferencing platform encompass the entire lifecycle regarding progress updates and setbacks. For example, the team leaders may host meetings to discuss task priorities and individual contributors may post their needs, and propose timelines for specific tasks and technical requirements, along with updates from their peer feedback. All of these project-related collaborations are often encapsulated within snippets, whiteboard sessions, instant messages, screen recordings, and in-channel document sharing. All of that data is either directly or indirectly connected to a project's backlogs.

To obtain explicit task sequencing (i.e., an item list) the summarization pipeline may include modality extraction and a sequence summarization model, as depicted in Figure 3, below.

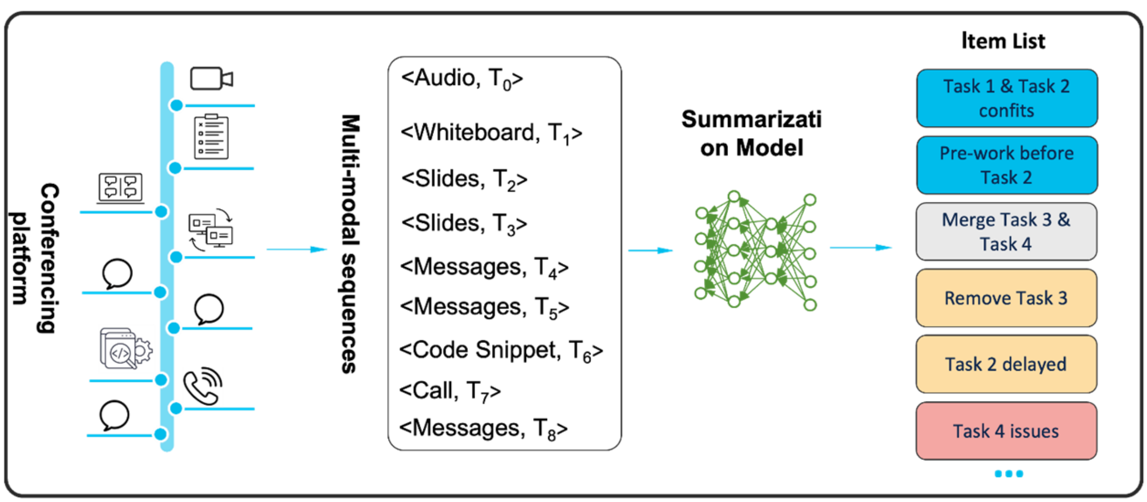

Figure 3: Multi-modal Project Summarization

As shown in Figure 3, above, the summarization pipeline serves to collect all of the knowledge to generate a sequential item list consisting of contributors' roles and the subtask descriptions across multiple modalities.

In a multi-modal project summarization module, the summarization model may transform a sequence of multi-modal project data (e.g., a video recording, a screenshot, a code snippet, an instant message, etc.) into a task-related sequence. In an output item list, the summarized topics reflect the task conflict, the task merging, and the dependent work that necessarily needs to be done.

In a modality extraction module, as shown in Figure 4, below, either state-of-the-art modality transformation models or fine-tuned deep learning models may work to extract information from the input sequence data on a conferencing platform.

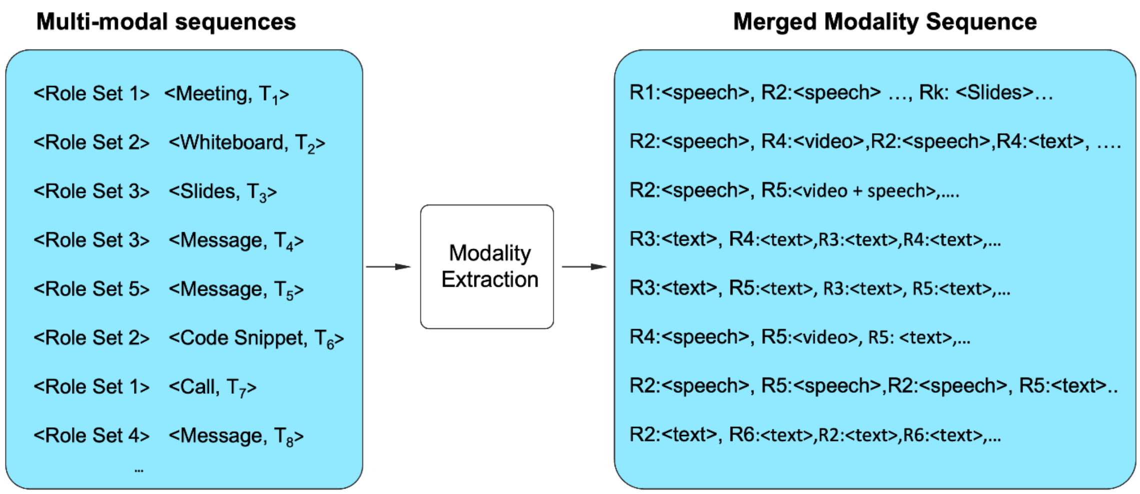

Figure 4: Multi-modal Sequence Extraction

For example, automatic speech recognition techniques may be utilized to transcribe video recordings or calls, and video retrieval techniques may also be able to slice a video frame and convert a resulting image to a description. The contributors (i.e., roles) may also be listed with the descriptions as the outputs. A multi-modal sequence including an audio recording, a whiteboard session, a screen recording, etc. may be utilized for task list generation. Specifically, the summarization model may focus on the task arrangement and descriptions with an understanding that the ground truth should be highly domain-specific.

As shown in Figure 5, below, a sequence summarization model may be utilized to summarize the descriptions that were obtained from merged modality sequences.

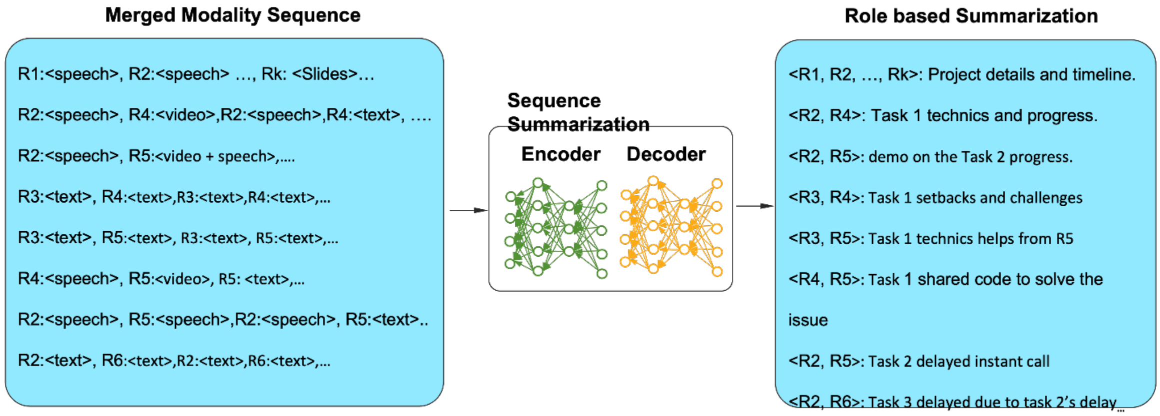

Figure 5: Multi-modal Sequence Summarization

Compared to general summarization models that focus on semantic context briefing, a project backlog-based summarization model, according to the techniques presented herein, needs to capture both local topics and the correlation of the global sequence to reveal task representations. Such an approach should be able to not only summarize the local context but also retain the memory of the overall sequence for an accurate representation of tasks. During this phase, the ground truth for the summarization model only focuses on the task sequences, and many task details are skipped and not included as training labels.

As noted above, a second key element of a MaPM solution (according to the techniques presented herein) encompasses an auto-agile model.

The output of a sequence summarization model, as presented above, roughly describes the task sequences that were obtained from a conferencing platform. As shown in Figure 6, below, a role-based summarization may wrap those collaborations as the detailed description plus the contributors involved.

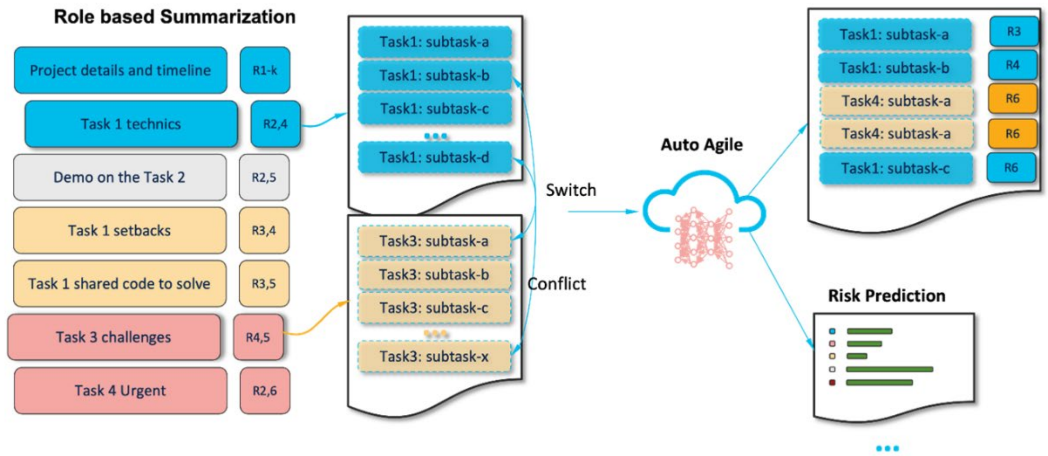

Figure 6: Auto-agile Task Generation Based on AI Techniques

In the detailed descriptions, subtask-related technical issues, progress, challenges, and setbacks may be summarized. However, logically, the task management-related merge, insertion, deletion, and priority-based shuffling cannot be directly revealed. Accordingly, an auto-agile model may be leveraged to project the role-based summarizations into a detailed list, in which the subtasks may be rearranged to avoid the conflicts, delays, and changes that were described in the summarizations. Additionally, the risk prediction for each subtask may be generated according to the overlapping timeline, the resource competitions, claimed challenges, and other factors among the different subtasks.

An auto-agile model, as described above, basically generates a task list to reflect all of the changes and task operations. For instance, the conflicts between subtasks, lagging behind the schedule caused by derailing off the main course, and the latest updated requests from a customer may be reflected by merging, splitting, and removing operations in the generated task list. In other words, the auto-agile model may extract the insights from the item list and then develop a new task list to avoid those issues and conflicts from the full-modality collaboration data.

The output of an auto-agile model, as described and illustrated above, encompasses a detailed task list and the corresponding risk predictions.

One of the most significant factors in agile project management is risk management. Due to the fast project delivery and short task lifecycle, the risk of failure can be one of the largest uncertainties in the product delivery. Since risk management is quite time sensitive, an auto-agile model (according to the techniques presented herein) provides an almost real-time risk prediction approach, which allows a project's owners to detect any risks in the first place and correspondingly maximize the chance to make changes to avoid those potential failures.

As shown in Figure 7, below, an auto-agile model may predict the risks for each subtask in the summarization sequence.

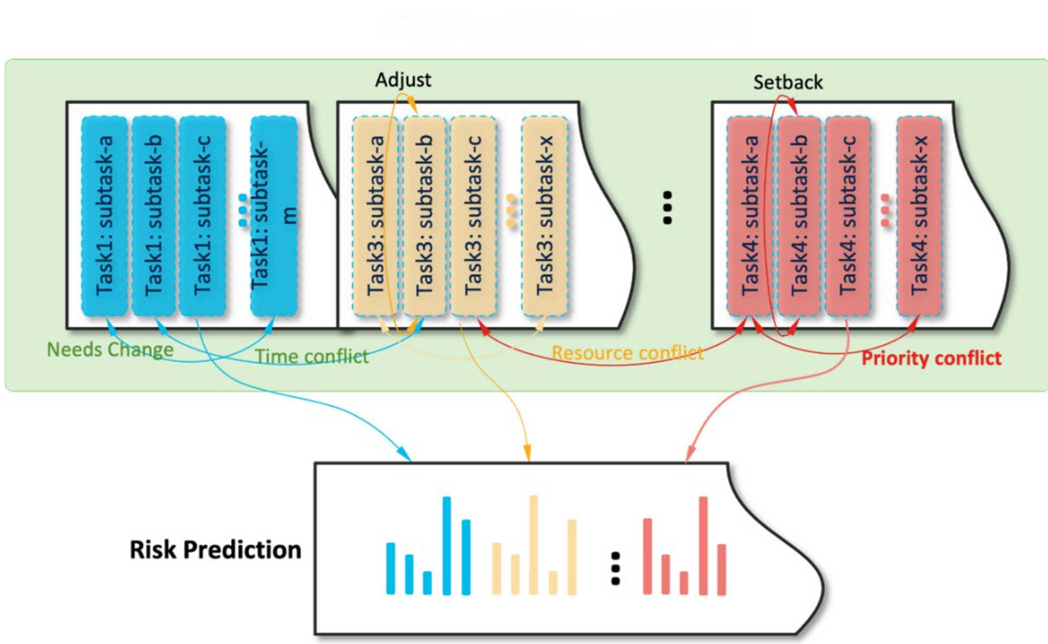

Figure 7: Subtask Risk Prediction in a Summarization Sequence

By leveraging a role-based summarization, the risk evaluation can comprehensively incorporate the full context for each individual subtask. An auto-agile model may detect the potential risks for each generated item. Unlike the general agile tools where each contributor estimates the risks of a task to their best knowledge, an auto-agile model can fully utilize the most comprehensive context information in the collaboration data to accurately assess the potential risks that usually cannot be fully appreciated by each individual contributor.

As noted above, a third key element of a MaPM solution (according to the techniques presented herein) encompasses agile sprint backlog optimization.

Utilizing the semantic contexts from a conferencing platform allows the techniques presented herein to fully consider all of the potential conflicts and setbacks in the sprint backlogs. However, the overwhelming details that are provided by the conferencing platform can result in task lists that cross over each other's priority. In particular, the risk estimation for each individual task may rely on the task sequences that were extracted from the conferencing platform, meaning that the impact of any necessary posterior operations (e.g., adding, deleting, merging, etc.) on the task lists is not considered in the risk estimation formula. For instance, certain tasks may need to be removed due to conflicts or challenges in completion, and, accordingly, the risks associated with those tasks should be emphasized. As a result, the generated task lists and risk estimations may, to some extent, include undesirable backlogs.

After an auto-agile model generates the detailed task list and corresponding risks, the techniques presented herein may further leverage a sprint backlog optimization module to regroup the task sequence and recommend a tailored sprint backlog. By regrouping a resource redistribution, the optimization module may strengthen the backlog generation and thus reduce the risk on each subtask. Figure 8, below, presents elements of such an approach.

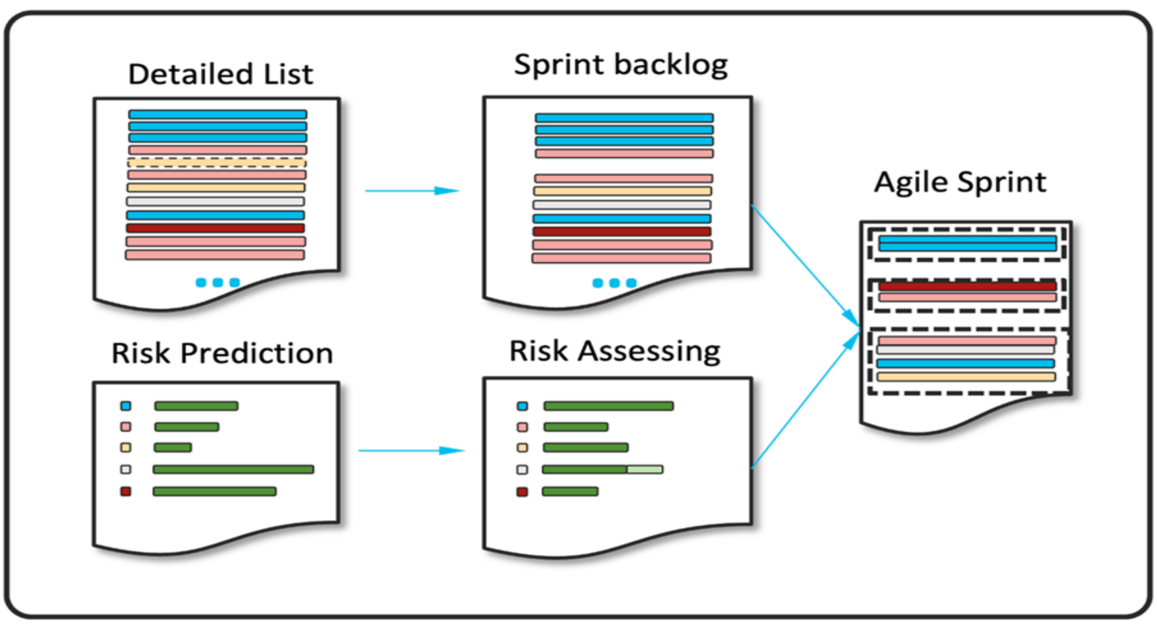

Figure 8: Agile Sprint Backlog Based on Predicted Risks and Task Reorganizing

As depicted in Figure 8, above, the general pipeline encompasses jointly passing the detailed list and the predicted risk scores to an optimization module where that module may generate an updated sprint backlog to reduce the overall risks among all of the subtasks.

For purposes of illustration, consider the operation of an optimization module, as described above, on an exemplary detailed list as depicted in Figure 9, below.

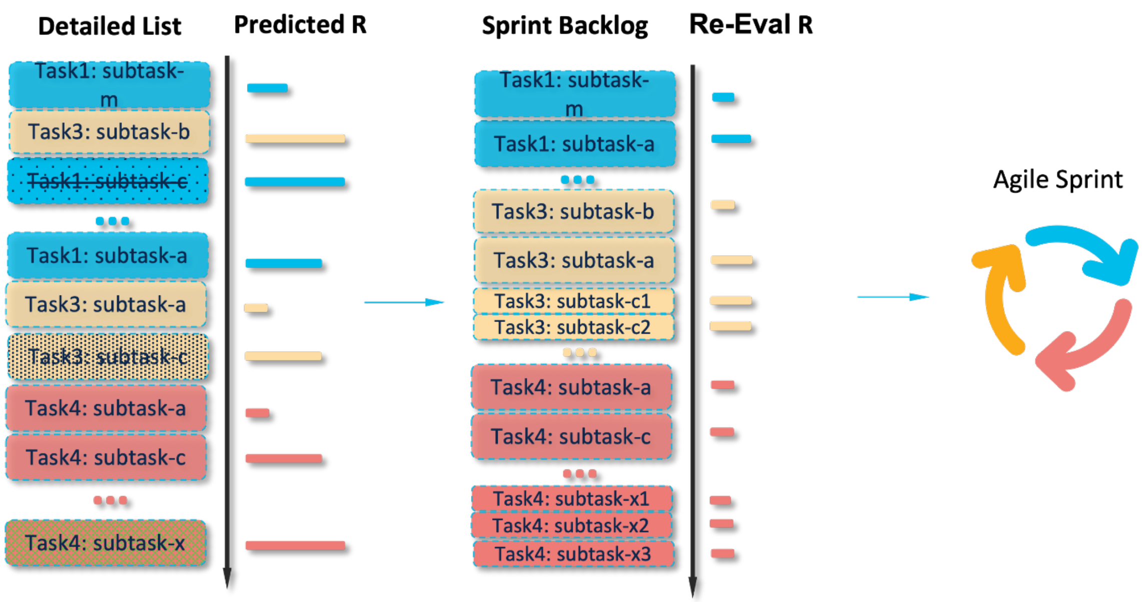

Figure 9: Optimization Module Operating on Exemplary Detailed List

As shown in Figure 9, above, subtask-c in task 3 and subtask-x in task 4 both need more resources based on the summarization from the collaboration data, and in the meantime subtask-c in task1 is incurring a very high risk. Following optimization, subtask-c in task 1will be removed from the list and, accordingly, more resources are added to both subtask-c in task 3 and subtask-x in task 4. By allocating more resources to those two subtasks, the relative risks are significantly reduced. To further fine tune the sprint backlogs, the optimization module may combine the generated risk scoring and task list to jointly implement refinements on the sprint backlogs. For example, tasks with very high risks may be removed and parallel tasks may be prioritized to reduce overall risks. Further, tasks with overlapping subtasks may be merged to select the subtask with a higher risk score.

The task regrouping may be accompanied through a re-estimation of risk scores, as the adjustments in task lists can affect the risks that are associated with each individual task and its preceding and following peers. Following the joint optimization of risk assessment and sprint backlog refinement, the final agile sprint may be regrouped to gather subtasks and finalize the sprint backlogs for successful agile project management. Overall, the generated sprint backlog has a lower risk compared to the detailed list.

In summary, techniques have been presented herein, which may be referred to herein as a MaPM system, that leverage a conferencing platform to offer a real-time agile project risk management and prediction framework that utilizes multi-modal collaboration data sources. Aspects of the presented techniques encompass an AI-backed summarization model that may be utilized to extract project details and an auto-agile that model may consume those loggings and generate the predicted project sprint backlogs and their risks. Further aspects of the presented techniques support an optimization module that may jointly update the predicted sprint backlogs and the estimated task risks to realize the finalized backlog sequences. In summary, an MaPM system, according to presented techniques, offers four novelties compared with conventional agile project management tools – a conferencing platform-centralized solution for automatic agile project risk prediction and management, a real-time multi-modal-based project risk monitoring and prediction system, the generation of sprint backlogs based on fully evaluated contexts that are collected from all of a project's participants, and the liberation of project contributors from having to conduct manual project tracking and recording.